

The Role of Academic Institutions in Validation and Verification Education

Paul Gray

Department of Computer Science

University of Northern Iowa

From an undergraduate, liberal arts perspective

- My perspective is going to slant toward computer science and numerical analysis.
- A significant omission in the support of V&V education is a well-defined cross-disciplinary program.
- High Performance Computing slant, for code that *works*.
- *In some ways, this perspective isn't "fair"*

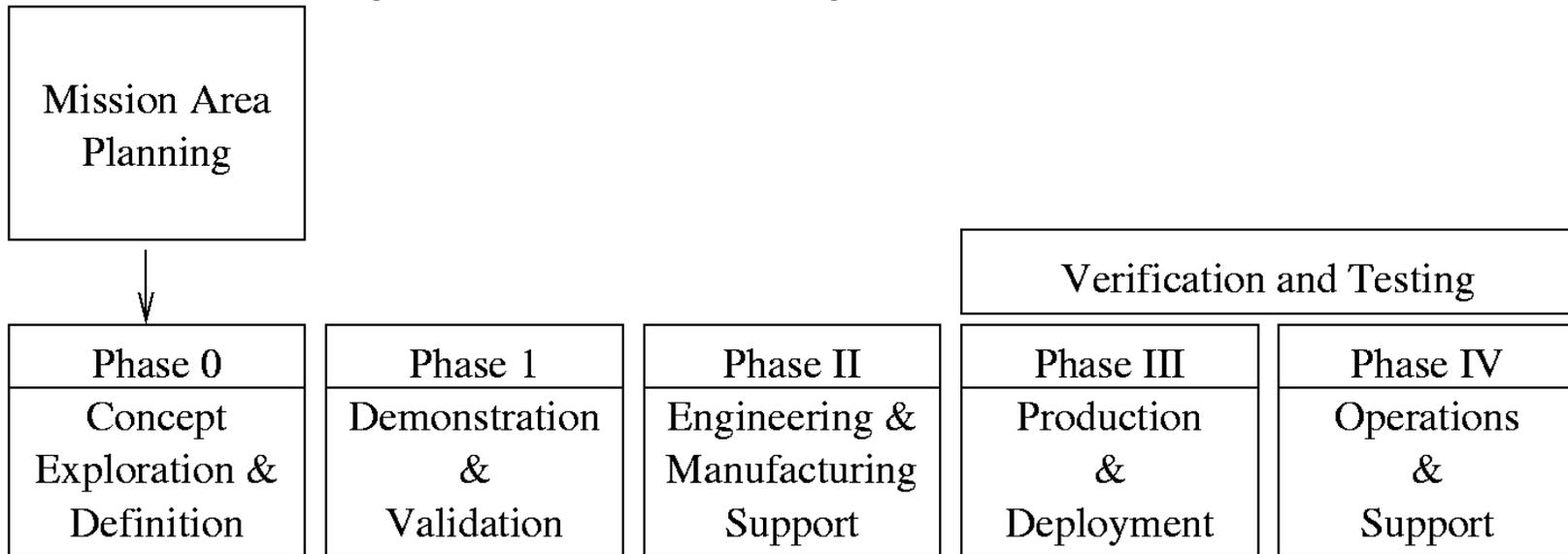
Where is V&V Today?

- Government
- Industry
- Academics?

Government and Industry are leaps and bounds ahead of Academics.

V&V in Government

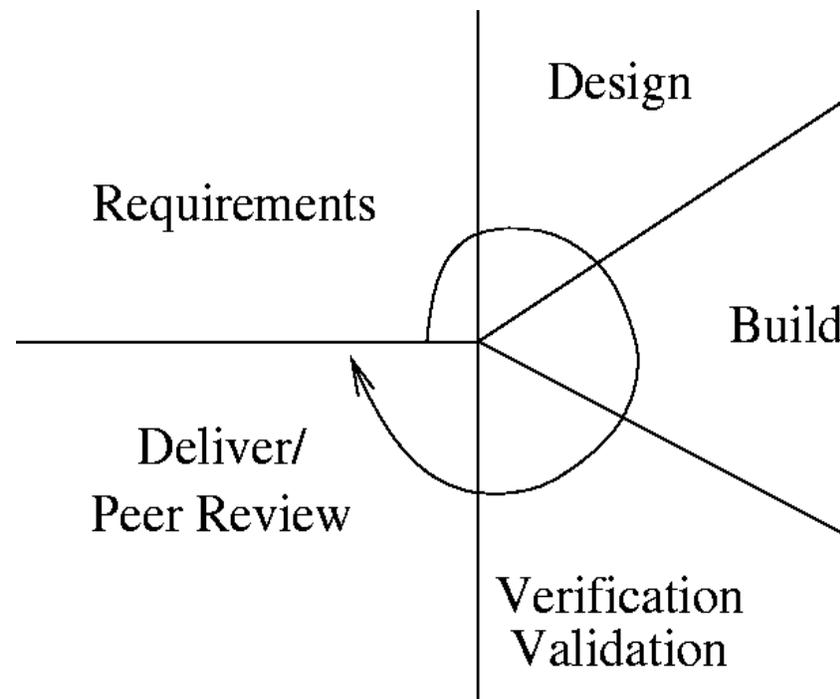
➤ Linear life cycle of DOD Systems



- Milestone-driven "waterfall" process
- System requirements are well understood and stable
- M&S supports every phase of the basic life cycle for DOD systems

V&V in Industry

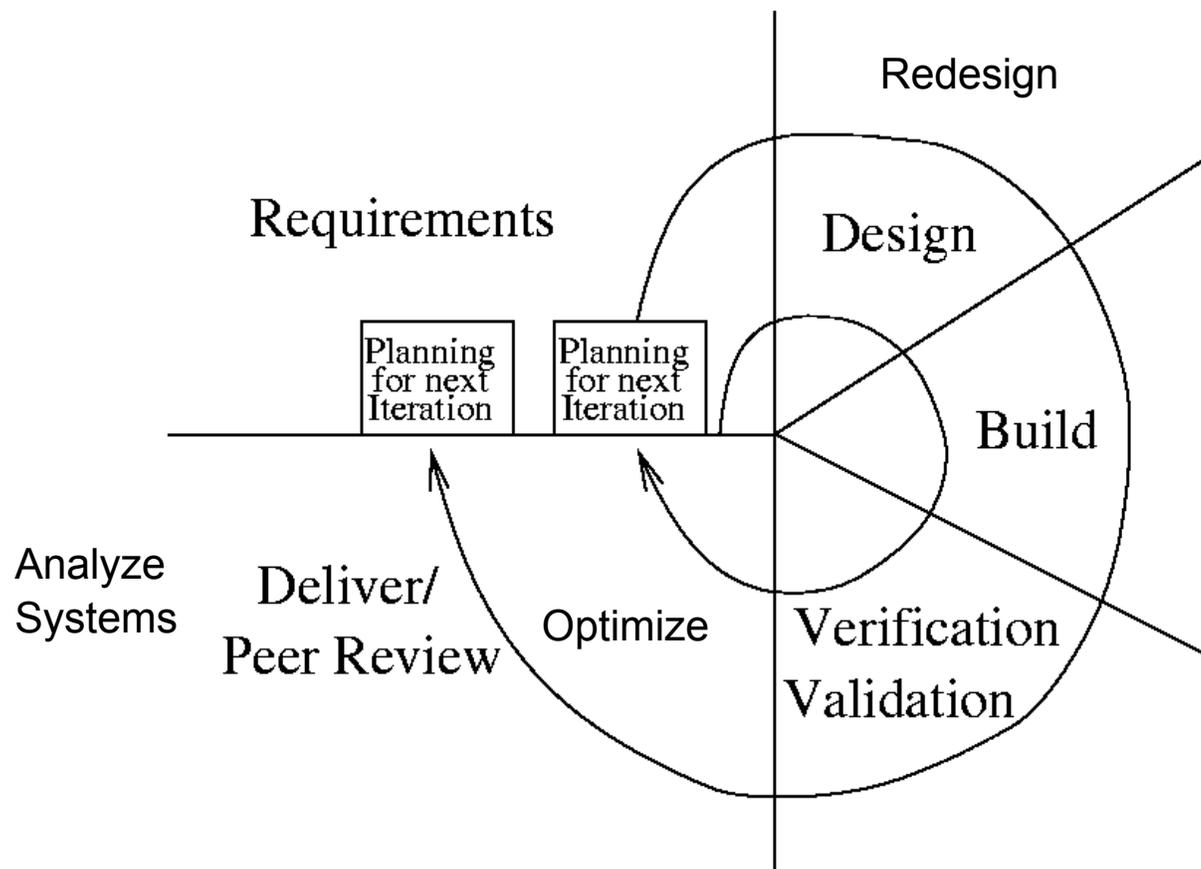
- Spiral development Cycle in Systems Eng.
 - Rapid Application Development Spiral



- Compare this to the model of service pack "security" bug/feature fixes and updates as part of the development cycle.

V&V in Industry

- Evolutionary development Cycle in Systems Eng.
 - Long-term refinement model



V&V in Education

- Lacking --- Academic participation at F02: 25%
- Engineering programs (Pace, ICSEE2K)
 - x California University systems
 - x Carnegie Mellon (Software Engineering Institute)
 - x Clemson
 - x Georgia Tech
 - x Johns Hopkins
 - x MIT
 - x Purdue
 - x Stanford
 - x Univ. of Central FL (Institute for Simulation and Training)
 - x Virginia Tech

V&V in Education

- How to convey development and documentation for M&S/V&V?
- V&V and Education are fundamentally hard to align:
 - V&V assumes a somewhat linear project cycle.
 - Education is realized as a topics-driven deliverable.
- Buchanan and Shortliffe (1987):
 - Nine different validation processes in the development process.
 - Prototypes, evaluations, refinements...
- How can academia introduce such rigor?

What about Balci's suggestions (and suggestions from other authors --- [e.g. Miller 92])?

Informal

Audit
Desk checking
Face validation
Reviews
Turing test
Walkthroughs

Symbolic

Cause-effect graphing
Partition analysis
Path analysis
Symbolic execution

Constraint

Assertion checking
Boundary analysis
Inductive assertion

Dynamic

Black-box testing
Bottom-up testing
Debugging
Execution monitoring
Execution profiling
Execution tracing
Field testing
Graphical comparisons
Predictive validation
Regression testing
Sensitivity analysis
Statistical techniques
Stress testing
Submodel testing
Symbolic debugging
Top-down testing
Visualizing
White-box testing

Static

Consistency checking
Data flow analysis
Graph-based analysis
Semantic analysis
Structural analysis
Syntax analysis

Formal

Induction
Inference
Lambda calculus
Logical deduction
Predicate calculus
Predicate transformation
Proof of correctness

Other bits and pieces: IEEE on Validation Testing

- *IEEE Standard for Software Test Documentation (1994) [8 points, summarized]*
 - *Plan for test activities in the development process. If the model is iterative, most test planning is done in the elaboration phase.*
 - *Design the tests. Devote time and resources during the development process to construct useful tests of the system.*
 - *Execute the tests. Test both at the unit level and at the overall system level.*
- *Are these testing activities present in the curriculum?*

Contrast to Academia Validation Testing

- Did it run? (Y/n)
- Did it SEGFAULT? (y/N)
- Is it due today? (Y/n)
- Okay, so it's not *that bad*, *but do we address:*
 - Does it have unsafe code usage?
 - How do we test for memory leads?
 - How do you test for use of uninitialized values?
 - Does it validate input?
 - How do we test for valid input?
 - What are the dangers of invalidated input?
 - Are the system outputs correct?
 - How do you tell?
 - How does it relate back to the model?

Standards: Bits and Pieces

- IEEE/ACM Standards provide minimal guidance for V&V.

- SE6 --- Software Validation [core]

- Course recommendation reads:

"Learning objectives:

- *Distinguish between program validation and verification.*
- *Describe the role that tools can play in the validation of software.*
- *Distinguish between the different types and levels of testing (unit, integration, systems and acceptance) for medium-size software products.*
- *Create, evaluate, and implement a test plan for a medium-sized code segment.*
- *Undertake, as part of a team activity, an inspection of a medium-size code segment.*
- *Discuss the issues involving the testing of object-oriented software.*

Standards: Bits and Pieces

- IEEE/ACM Standards provide minimal guidance for V&V.
 - SE10 --- elective, Formal Methods (a hidden "gem")
 - Course recommendation reads:
"Learning objectives:
 - *Apply formal verification techniques to software segments with low complexity.*
 - *Discuss the role of formal verification techniques on the context of software validation and testing*
 - *Explain the potential benefits and drawbacks of using formal specification languages.*
 - *Create and evaluate pre- and post-assertions for a variety of situations ranging from simple through complex.*
 - *Using a common formal specification language, formulate the specification of a simple software system and demonstrate the benefits from a quality perspective.*

Standards: Bits and Pieces

- IEEE/ACM Standards provide minimal guidance for V&V.

- SE11 --- elective, Software Reliability

- Course recommendation reads:

- "Learning objectives:*

- *Demonstrate the ability to apply multiple methods to develop reliability estimates for a software system.*

- *Identify and apply redundancy and fault tolerance for a medium-sized application.*

- *Explain the problems that exist in achieving very high levels of reliability.*

- *Identify methods that will lead to the realization of a software architecture that achieves a specified reliability level.*

- *Topics:*

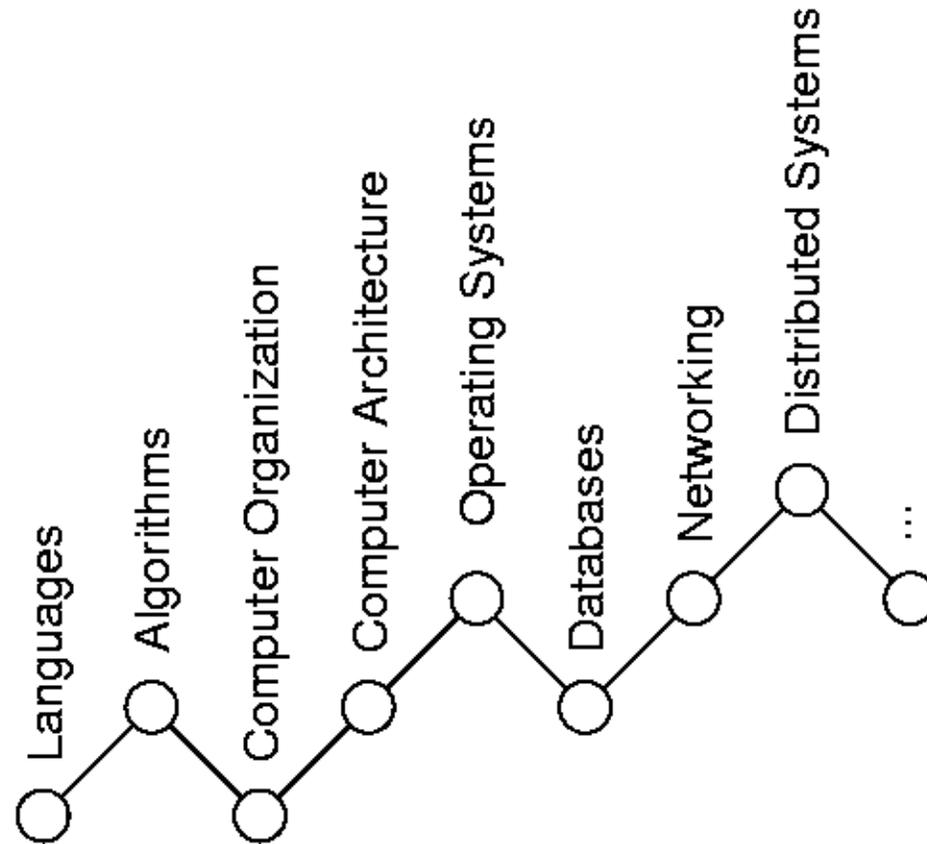
- *... Probabilistic methods of analysis*

Standards.

- What about other aspects of CS education?
 - Simulations on High Level Architectures (HLA's) push the limits of V&V
 - Networks of workstations
 - Distributed heterogeneous environments
 - Federated clusters (clusters of clusters)
- American Mathematical Society (AMS) Curricular review provides no references to V&V [Actually, to V or V!)
- American Physics Society (APS)?
- What about standards to bring together the standards?

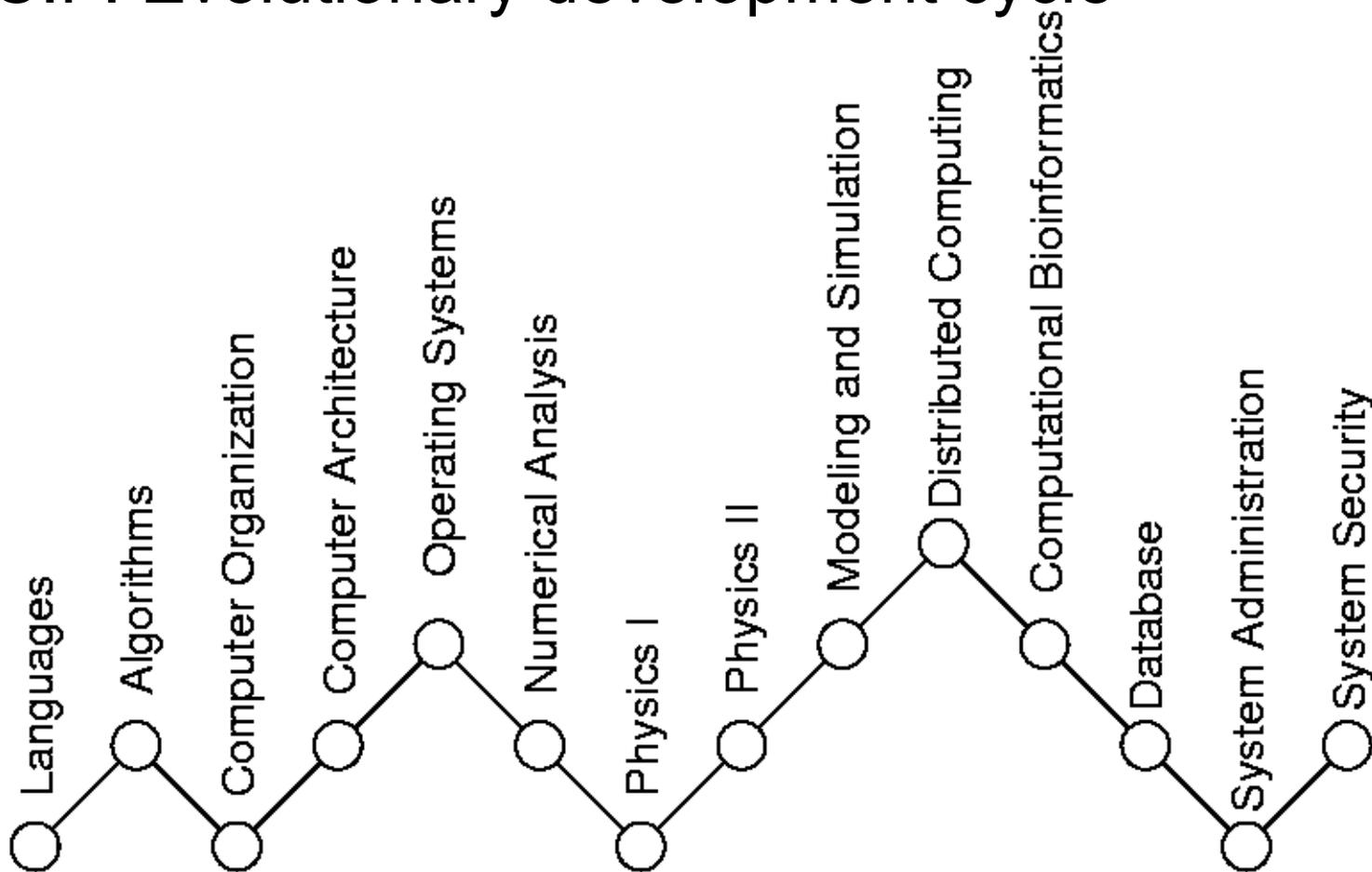
Linear Disciplines

- Most of the previous comments are about CS
- M&S and V&V involve strong, cross-disciplinary concepts

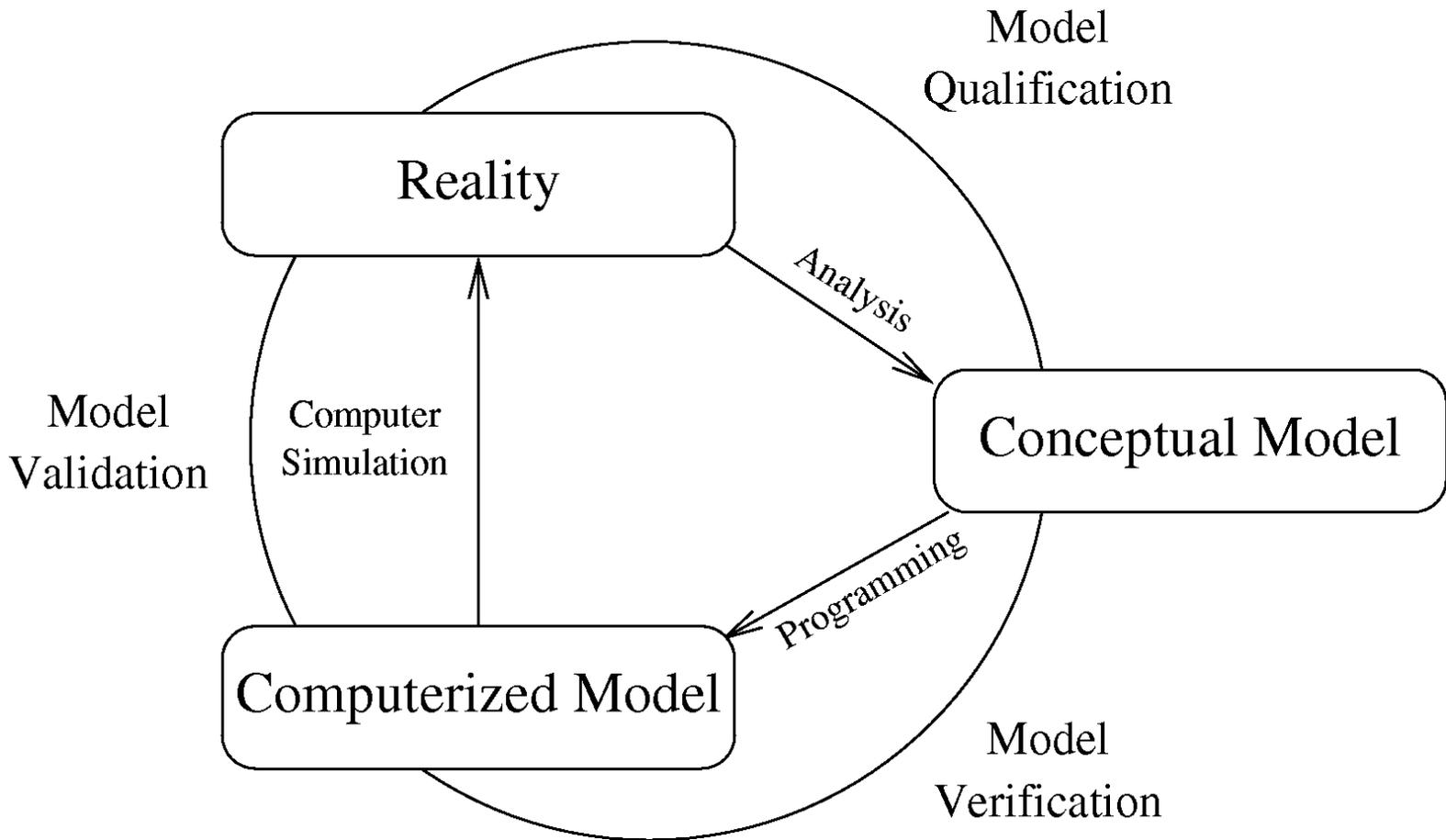


Multigrid Disciplines

- Cross-discipline paradigm shift required
- Paradigm shifts are a big-O operation
- C.F. Evolutionary development cycle

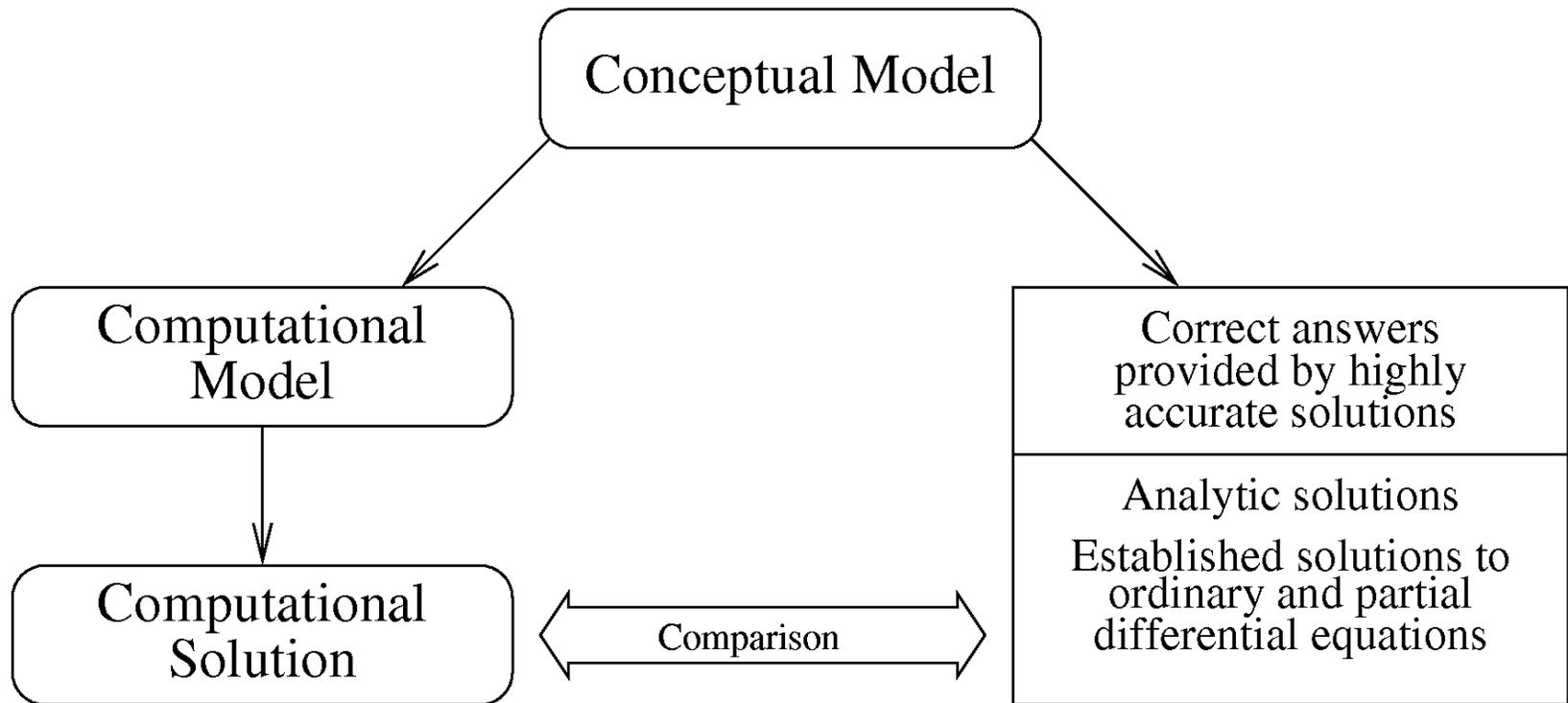


AIAA Model



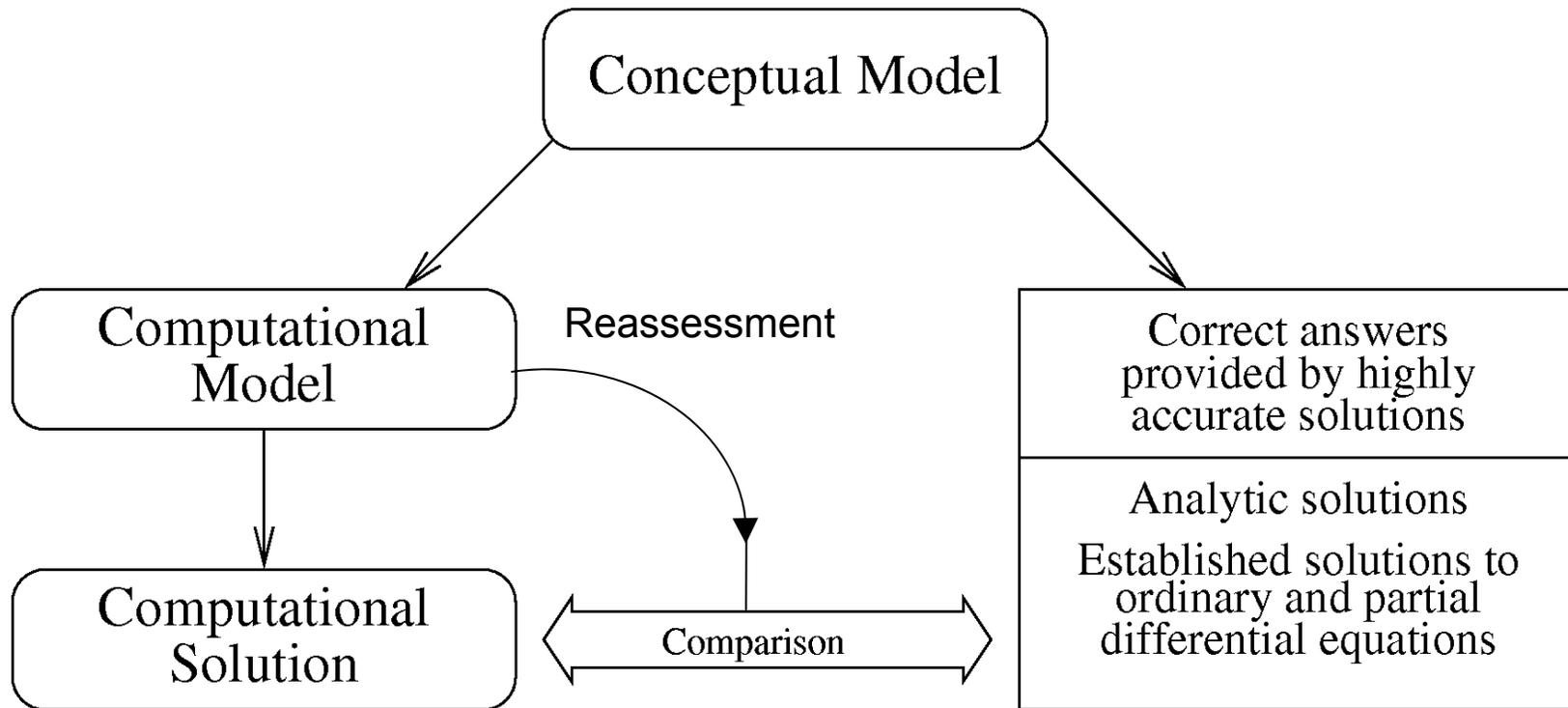
Where does education fit in?

AIAA Model ---- Verification



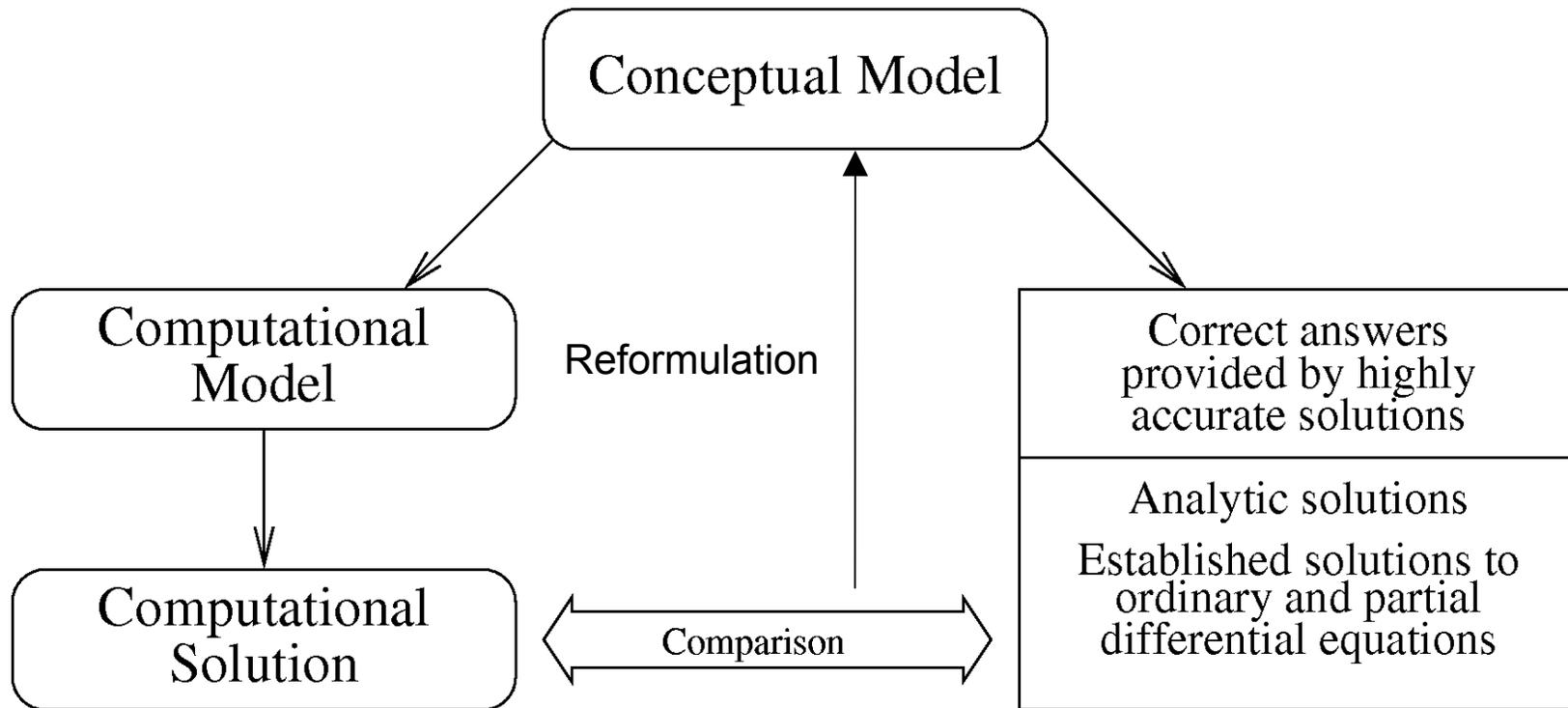
Where does education fit in?

AIAA Model ---- Verification



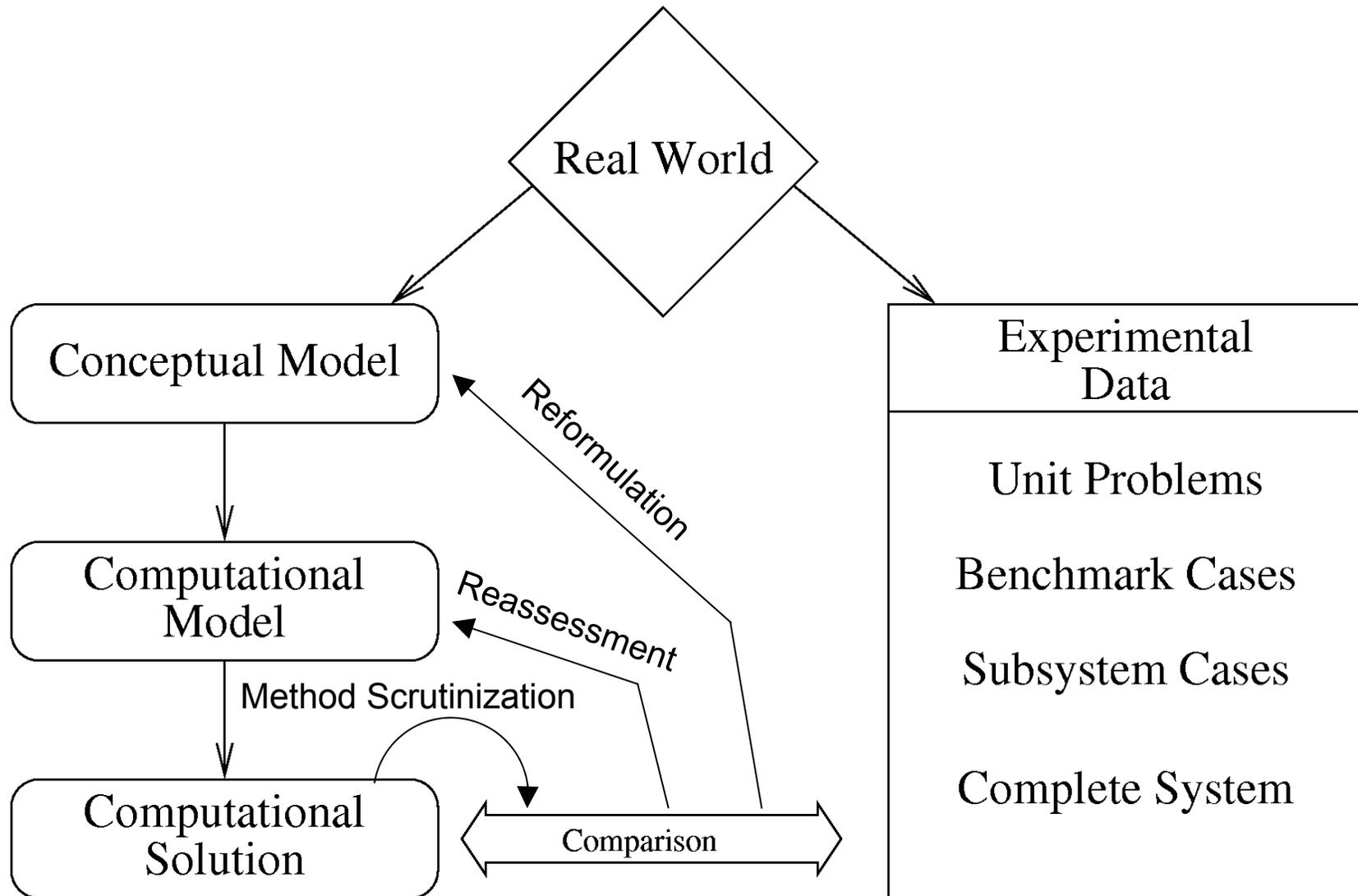
Address errors in the numerical approximation

AIAA Model ---- Verification



Address errors in the formation of the model

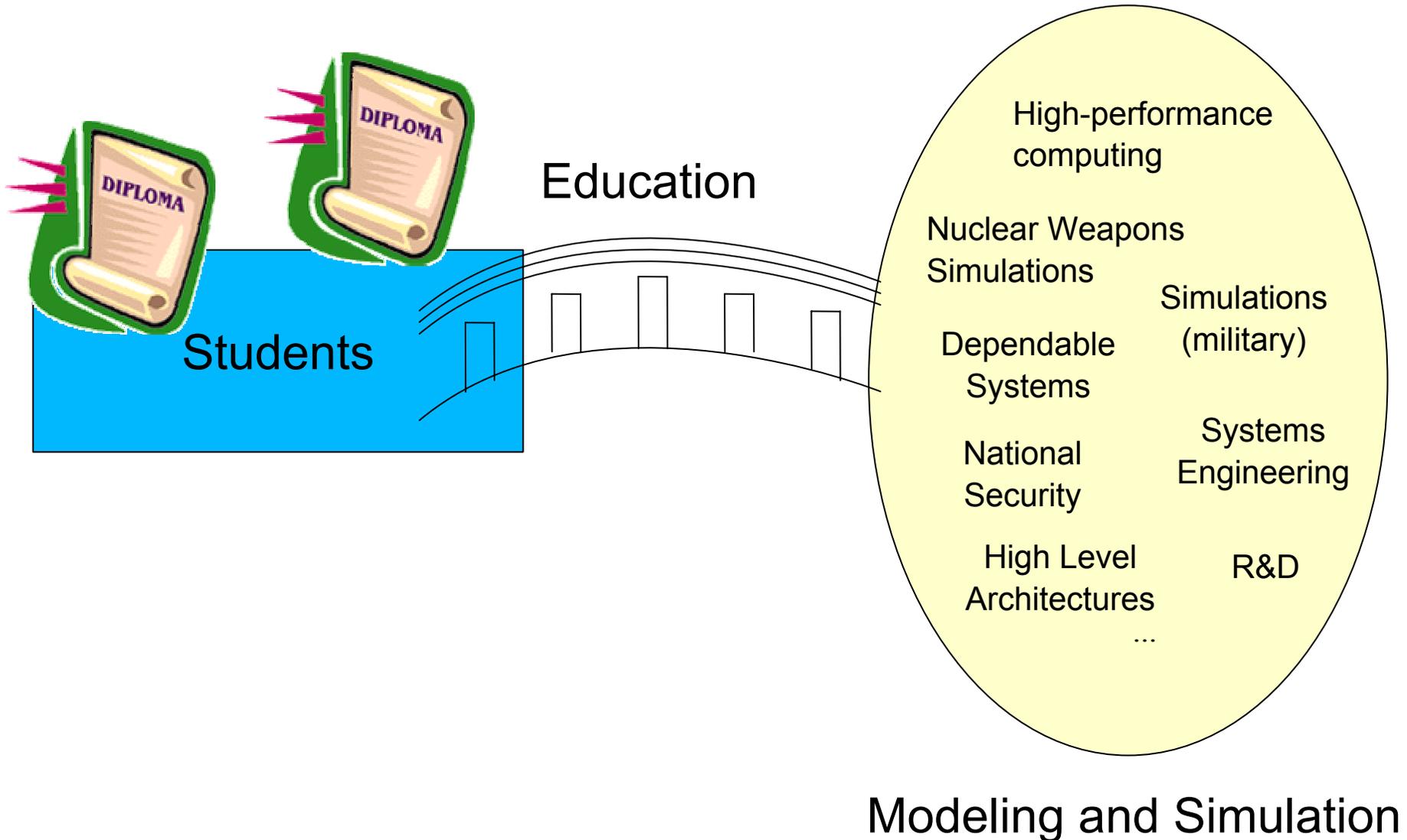
AIAA Model ---- Validation



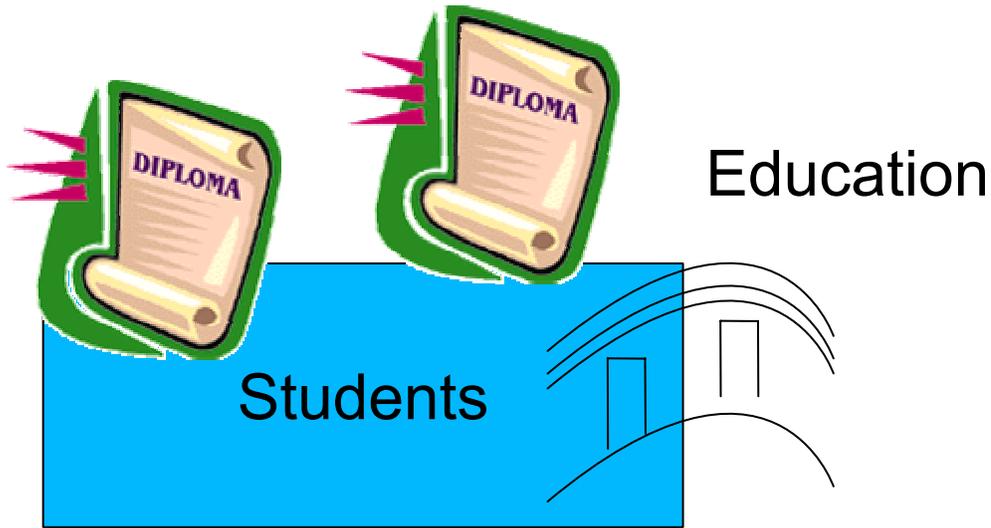
Topics for which validation applies:

- Completeness
- Efficiency
- Validity
- **Maintainability**
- Consistency
- Precision
- Soundness
- Usability
- Justification
- Reliability
- Accommodating
- Clarity
- Quality

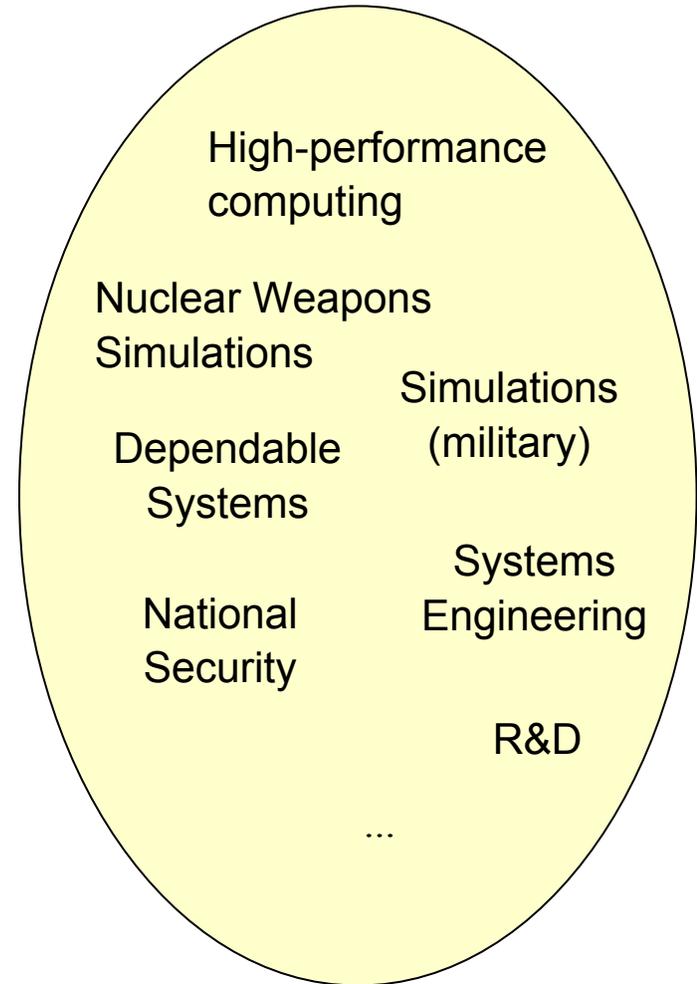
Education is the bridge



Education: V&V as a "Topics" Supplement

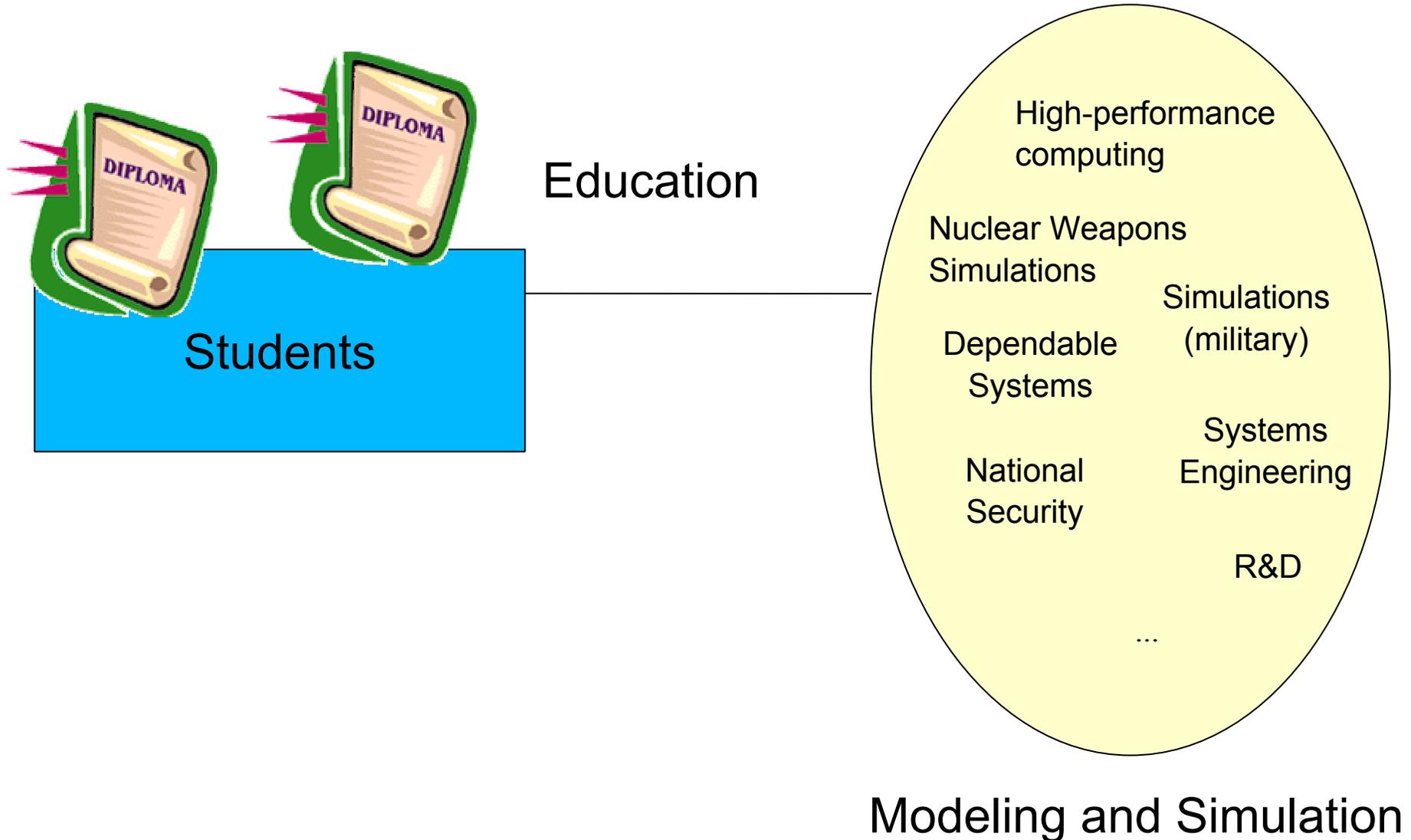


Depth, but no breadth

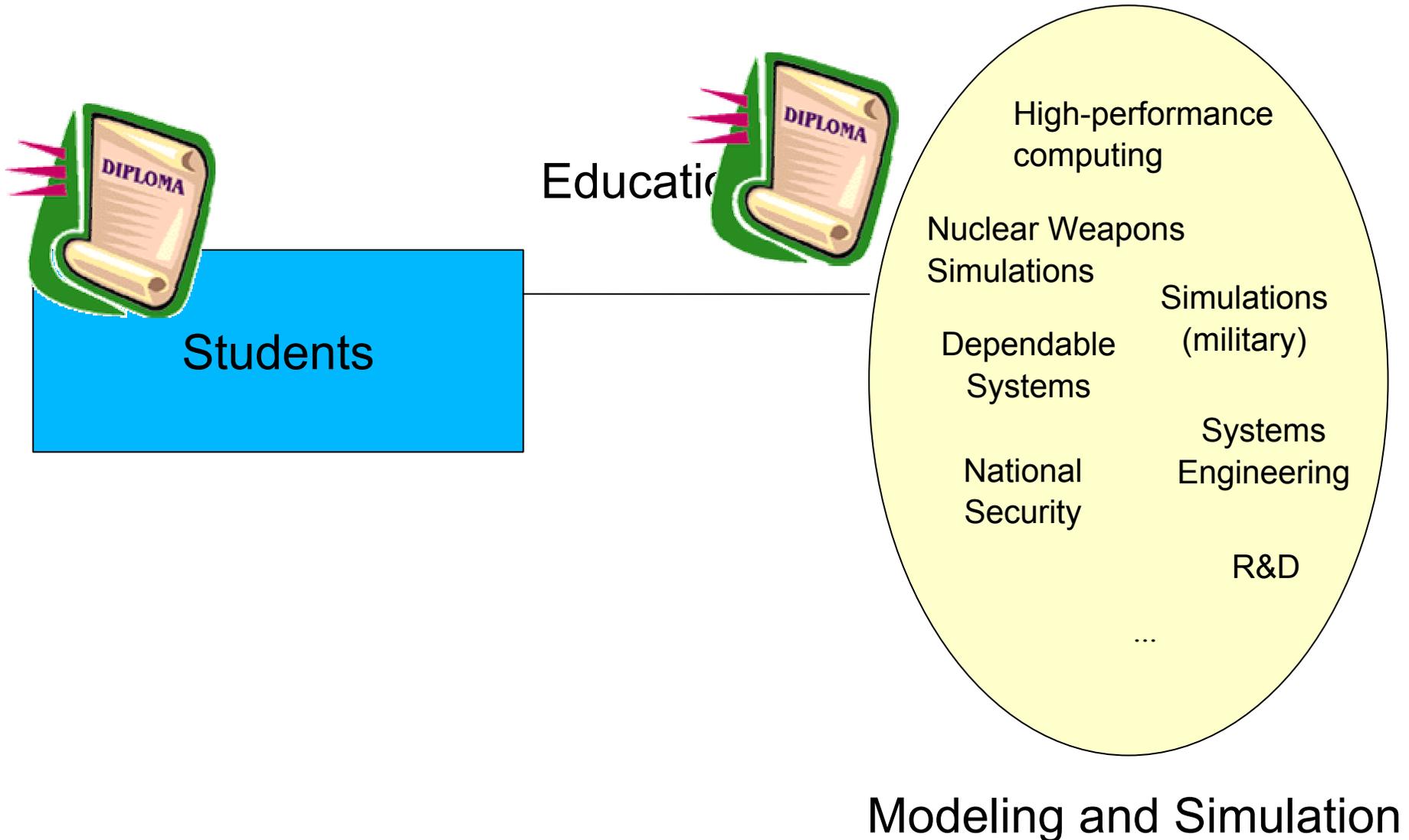


Modeling and Simulation

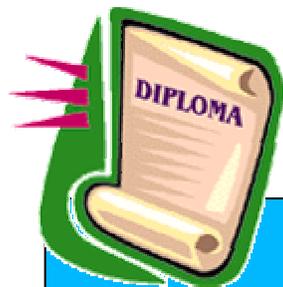
Education: V&V as a Project Culmination



Education: V&V as a Project Culmination



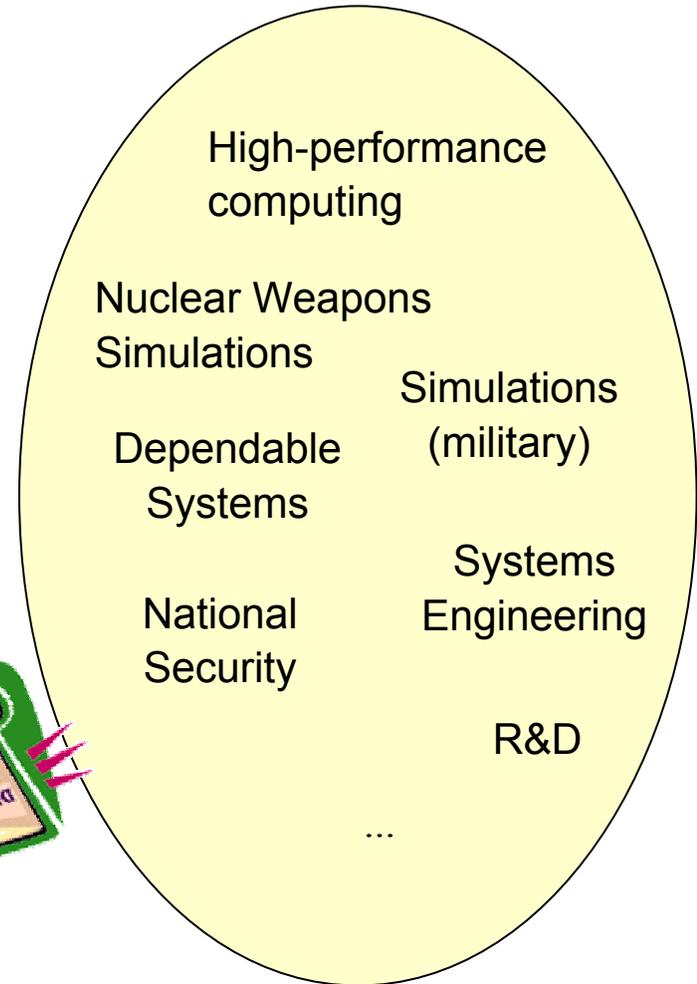
Education: V&V as a Project Culmination



Students

Education

Breadth, but no depth



Modeling and Simulation

Return on Investment

- Similar to Industrial and Govt.
- Academic institutions need to address ROI:
 - What is the "cost" of V&V education?
 - Added prep time
 - At the expense of other topics
 - How to integrate long-range concepts in "modules," "courses," and the curriculum?
 - What is the payout?
 - Far removed --- how does V&V improve returns on academic "product?"
 - Contrast with MC02, or NASA Mars explorer.
 - V&V does not drive academics like it does M&S projects.

Modules, Courses, and Programs

- Levels where you can integrate V&V
 - At the "module" level
 - Short topics
 - Could be as simple as revisiting the original model and asking the right questions.
 - At the "course" level
 - Appropriate for a dedicated "projects" course, with well-defined entry and exit criteria
 - Appropriate for a course in mathematical modeling
 - Appropriate for a course in mathematical physics
 - Appropriate for a course in software engineering
 - At the "program" level
 - What would a "certificate" in V&V entail?

Modules and courses

- Serve to reinforce the topic and the motivations.
 - Here is our mathematical derivation of the physical model. Here is our solution to the mathematical model. Let's examine if the solution reflects reality.
- Requires cross-disciplinary courses that will address:
 - A good deal of statistics
 - Use (and understanding of) differential equations
 - A tool chest of numerical methods
 - An understanding of stochastic analysis

Programs

- *A program that supports V&V must involve cross-disciplinary contact and instruction.*
- *Compare this with Pace's recommendation on Subject Matter Experts (SMEs).*
- *So where can academia start.....???*

A good start: Focus on the Education of *Errors*

- Because there are some things you just can't control --- but you can identify.
- ...but that doesn't mean that you can't detect and overcome errors in modeling, simulations, numerical/analytical solutions.
- Identify:
 - Errors in the equations of the model
 - Blunders, goofs, mistakes
 - Errors in safe coding
 - Errors in data collection
 - Numerical algorithms errors
- What happened to the once-active area of interval analysis?

A good start: Focus on the Education of *Errors*

- Conveys *thinking and intuition aspects*
- *Cross discipline*
- Errors in the equations of the model
 - Fundamental to understanding validation issues
- Blunders, goofs, mistakes
 - Algorithmic errors
- Errors in safe coding
 - Neglect of round off, critical sections, input validation
- Errors in data collection
 - Real-world data has inherent data-collection errors.
- Numerical algorithms errors
 - Rigorous analysis of algorithmic convergence (stiff!).

Textbooks

- To support V&V modules, courses, or programs, educators need the materials (right?)
- Are there textbooks that sufficiently support V&V aspects of M&S?
- Some Software Engineering texts do a good job on verification, but general presentations of V&V practices are just not found.
- Texts and supplements that focus on *academic pursuits of V&V are needed*

Items that should be covered:

- A great deal of statistics
- Use and understanding of ordinary and partial differential equations
- A tool chest of numerical methods, along with an thorough algorithmic analysis.
- An understanding of stochastic analysis
- Software design and testing

Resources

- Textbooks (academic) are needed.
- SME's <=-----=> Cross-disciplinary labs.

Where to insert V&V materials

- Error Investigations
- As iterative refinements to topics

Conclusions

- The greatest ROI for V&V education is simple reinforcement of V&V principles as curricular "modules"
- Focusing on errors in modeling would also go a long way to supporting *V&V thinking*.
- To develop V&V courses or programs, strong interdisciplinary components will be required.
- Requires:
 - A good deal of statistics
 - Use (and understanding of) differential equations
 - A tool chest of numerical methods
 - An understanding of stochastic analysis

Conclusions

- The greatest ROI for V&V education is simple reinforcement of V&V principles as curricular "modules"
- Focusing on errors in modeling would also go a long way to supporting *V&V thinking*.
- To develop V&V courses or programs, strong interdisciplinary components will be required.
- Requires:
 - A good deal of statistics
 - Use (and understanding of) differential equations
 - A tool chest of numerical methods
 - An understanding of stochastic analysis