Unifying Verification and Validation of Software Systems

Content

- **Methods for Verification and Validation**
- **Current Research and Teaching**
- **Why a unification and a class on unification?**
- **Elements of a unification**
- **Course approach**
- **Why take this course?**

Verification and Validation

Specification of the properties that a software system must have and determination that the system possesses these properties.

Property

- **“The value of x will never exceed 13.”**
- **“The value of n input to sort(n,a) will never exceed 200.”**
- **“It will never be the case that process P is in state A and process Q is in state B.”**
- **“A value from a data element with security level k will never be assigned to data element with a lesser security level.”**

Informal Definitions of Five V&V Methods

- *Testing* – Determines the correctness of the execution of a program for a given initial condition and input set.
- *Static Analysis* – Determines program properties such as data-flow paths and control flow paths that can be deduced from the static structure of the program.
- *Model Checking* – Determines the correctness of a temporal property for the executions of a program for all initial conditions and inputs.
- *Formal Proof* – Determines whether a program conforms to a specification of behavior, usually an input/output relation for all executions of the program.
- *Runtime Monitoring* – Dynamically checks whether the execution of a program conforms to a specified condition.

Current State of Research and Teaching in Verification and Validation

Research

Separate research communities in each of static analysis, testing, model checking, theorem proving and runtime monitoring.

Each has its own group, its own vocabulary and its own professional meetings.

Teaching

Little coverage of design methods for verifiability

Each is typically taught separately from one another

Unification of Verification and Validation Methods

What is wrong with this?

1. **No single method is universally effective by itself**
2. **Each has strength and weaknesses in applicability**
3. **Synergisms among the methods abound**
4. **All are based on common underlying principles**
5. **Does not produce most effective V&V Process**

What should we be doing?

Formulating a coherent and comprehensive approach to verification and validation that begins with the design for verification and validation.

Teaching this unified approach at both undergraduate and graduate levels.

Unification of Verification and Validation Methods

Elements of a Unification

1. **Define a standard property specification language to be used for property specification regardless of the method of verification.**
2. **Establish a design and development process which yields artifacts which are amenable to validation and verification.**
3. **Establish a common basis for abstracting programs for verification.**
4. **Explore and exploit the synergisms among the methods**
5. **Map property/system pairs to appropriate mechanisms.**

Approach

- **Combine formal methods with practical applications**
- **Specification of properties – templates and tools**
- **Design and Implement for Verification**
- **Apply best of breed verification tools**

Why Take this Course??

- **Learn unique skills**
- **Add to employability**

  Mathew Mengerink – Head of architecture, systems, and technical strategy for PayPal.

  > **“I remember you jumping up on the table and shouting out to an entire lecture hall full of students, ‘If you do not design testability into the code from the beginning, you're a goddamn fool.’ Funny, the majority of our Stanford and MIT grads still don't quite get where unit tests fit in and how they help optimize development.”**
- **Participate in NSF Funded Project**