Workshop - Modeling Software the Alloy Way

Michael J. Lutz
Software Engineering Department
Rochester Institute of Technology
Rochester, NY 14623-5608, mjl@se.rit.edu

Abstract - Software engineering educators struggle to make formal design methods both accessible to and meaningful for students; the lack of appropriate engineering tools is especially problematic.

One tool that addresses these problems is Alloy from Daniel Jackson’s Software Design Group at MIT. Alloy has a formal mathematical foundation, making it useful for high-level system modeling, and for exploring the consequences of design decisions.

This workshop will focus on using Alloy to teach formal modeling. While the workshop will partly tutorial, the goal is to help computing educators develop the skills, knowledge, and motivation to use Alloy in software engineering classes.

Index Terms – Software engineering, formal methods, mathematical models of software.

OVERVIEW

A hallmark of engineering design is the use of models to explore the consequences of design decisions and to verify the properties of proposed designs. In contemporary engineering, this frequently means the use of mathematical models to aid such investigations. Unfortunately, most such tools for modeling software lag far behind those in traditional engineering, both in terms of usability and utility.

Until now, those who taught mathematical modeling (or “formal methods”) faced daunting challenges. First, most modeling tools used seemingly esoteric notations that were hurdles for many students. Even if the notation could be tamed, the tools themselves were rarely more than syntax checkers, possibly with support for simple expression evaluation. Venturing beyond this requires understanding of proof theories and strategies well beyond that typical of other engineering disciplines. What is more, the tools worked at a much lower level than that of the domain itself; it was easy for students to miss the forest for the trees.

The development and release of the Alloy from MIT [1] has improved the situation dramatically. With Alloy, instructors now have a tool that supports formal structural and behavioral modeling (using C-like syntax), along with state space exploration and property verification using relational logic, predicates, and assertions. The tradeoff involved – only first order systems over finite domains can be analyzed – is not problematic in practice.

WORKSHOP GOALS AND FORMAT

The goal of this workshop is to introduce Alloy – both the language and support tool [2] – to faculty interested in formal methods and mathematical modeling. After a brief introduction to Alloy concepts, the tool and language will be explored by interactively developing a simple software system model. This approach mirrors the way Alloy is taught and used within RIT’s undergraduate software engineering program [3],[4].

TOPIC OUTLINE

2. What Alloy’s Analyzer Can (and Cannot) Guarantee.
3. Alloy Foundations: Signatures, Atoms, Sets, Relations
4. Specifying constraints: Facts and Predicates
5. Relational navigation: Joins and Transitive Closures
6. Exploring the state space: The Run command
7. Verifying properties: Assertions
8. Modeling Operations: Adding Time
9. Putting It All Together: Completing The Model
10. Questions and Discussions

REFERENCES