

Software Model Checking Using Bogor

- a Modular and Extensible Model Checking Framework

*3rd Estonian Summer School in
Computer and System Science (ESSCaSS'04)*

Slide Set 00: Overview of Lectures

<http://bogor.projects.cis.ksu.edu>

<http://www.cis.ksu.edu/~hatcliff/ESSCaSS04>

John Hatcliff

Matthew B. Dwyer

Robby

SAnToS Laboratory, Kansas State University, USA

Support

US Army Research Office (ARO)
US National Science Foundation (NSF)
US Department of Defense
Advanced Research Projects Agency (DARPA)

Boeing
Honeywell Technology Center
IBM
Intel

Lockheed Martin
NASA Langley
Rockwell-Collins ATC
Sun Microsystems

Research Context

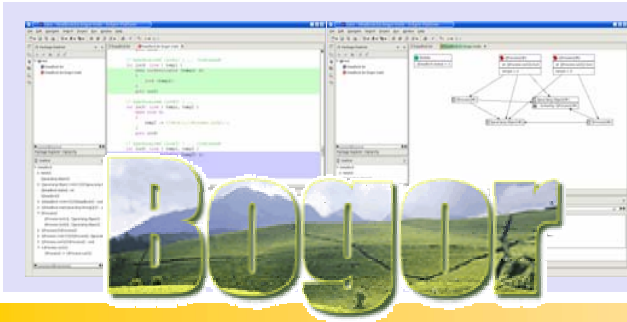


SAnToS Laboratory,
Kansas State University
<http://www.cis.ksu.edu/santos>

- Research on **Static Analysis** and **Transformation of Software**
 - model-checking, static analysis, model-driven component-based development, slicing, partial evaluation, etc.
- Aiming for robust tools
 - open source, close to commercial quality
- Integration into development process
 - ease of use and scalability sometimes take precedence over theoretical elegance
 - most of the time, focus is on bug-finding rather than true verification
- Trying to build on lessons learned...
 - ...from previous versions of tools (e.g., old Bandera)
 - ...from interaction with industrial partners

Research Context -- Bogor

In ESSCaSS'04 lectures, I'll focus on...



Bogor Model Checking Framework
<http://bogor.projects.cis.ksu.edu>

- Supporting model-checking of OO software (Java, in particular)
- Open platform for research/experimentation
 - take your favorite new idea, implement it in Bogor to try it out
- Teaching tool
 - foundation of a tool/application-oriented course on model-checking
 - some material already available; much more on the way

Perspective for ESSCaSS'04

- There is a lot of great stuff going on in the area of program verification and software model-checking
- I'll focus on work that our group has done
 - ...you can look in papers referenced as background reading for related work from our group and others
- You'll get my perspective on software model-checking
 - ...there are many other valid perspectives
- Lecture themes
 - Bogor as an extensible model-checking framework
 - Using Bogor to build your own customized model-checker
 - The value of domain-specific model-checking customizations
- Lecture non-themes (very important, but not covered)
 - automata-theoretic view of algorithms, symbolic model-checking (BDDs), details of temporal logic, data/predicate abstraction

Lecture Outline

- Lecture 1: Foundations of Model-Checking
 - Overview of Bogor
 - Overview of basic depth-first search explicit state reachability algorithm
- Lecture 2: Bogor Architecture & Extensions
 - Toward understanding the guts of Bogor and how to modify it
 - How to extend Bogor's modeling language
- Lecture 3: Representing Java, Checking Specifications
 - Representing Java in Bogor
 - A quick overview of Bogor reduction algorithms
 - Checking JML specifications
 - Checking atomicity specifications
- Lecture 4: Customizing Bogor to Check Avionics Designs
 - Cadena – an IDE for design of component-based distributed systems
 - Checking Cadena designs in Bogor

Tools Used in Lecture

- Bogor installation
 - demos during lectures
 - on your CD, includes user manual, examples, etc.
 - *if you use Bogor, we would like you to register on the Bogor web-site!*
 - *<http://bogor.projects.cis.ksu.edu>*
- Cadena
 - demos during lectures
 - available for download

Supporting Material

- Chapter from our lecture notes of foundations of model-checking
 - in your printed material, on CD
- Tutorial on Writing Bogor Extensions
 - in your printed material, on CD
- Research papers for background reading
 - on CD
- *All supporting material on CD can also be found at the following site...*
 - <http://www.cis.ksu.edu/~hatcliff/ESSCaSS04>

Other SAnToS Tools & Material

See your printed hand-out for more information...

- Bandera
 - Java Model-checking Environment
 - next generation due out an end of 2004
- Cadena
 - environment of design, specification, and analysis of distributed component-based systems
- Indus
 - static analysis and program transformation tools (e.g., full Java slicer)
- Course material on software specifications
- Course material on foundations and applications of model-checking