A Graduate Seminar in Tools and Techniques

Patrick J. Graydon
M. Anthony Aiello
Elisabeth A. Strunk
John C. Knight

University of Virginia

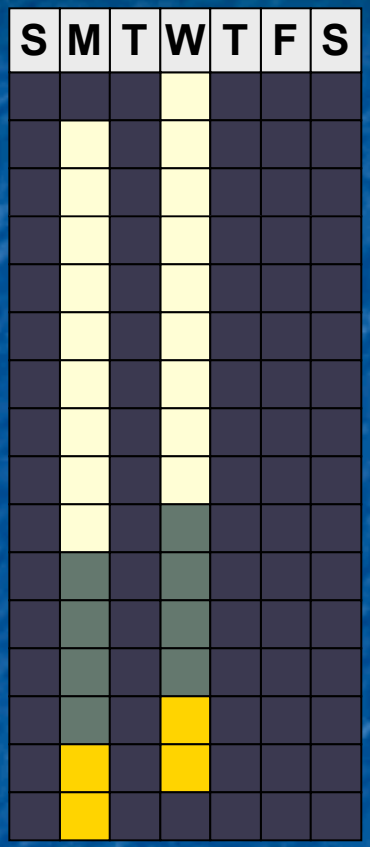
A Seminar...

- For SE grad students
  - Particularly in *dependable systems*
- To introduce, not build proficiency
  - Introduce tools and concepts
  - Students critique the tools and concepts
- On model checking and model-based development
  - They're both important
  - Because we can

Tools and Techniques

- Each student picked a tool:
  - **SLAM** - driver model checker from Microsoft
  - **BLAST** - model checker from UC Berkeley
  - **Kronos** - model checker from Verimag
  - **Spin** - Model checker developed at Bell Labs
  - ***SCRtool*** *- Specification tool from NRL*
  - **PerfectDeveloper** - design-by-contract development tool from Escher Technologies
  - **SCADE** - MDB tool from Esterel Technologies
  - **Simulink** - MDB tool from The MathWorks

Course Organization

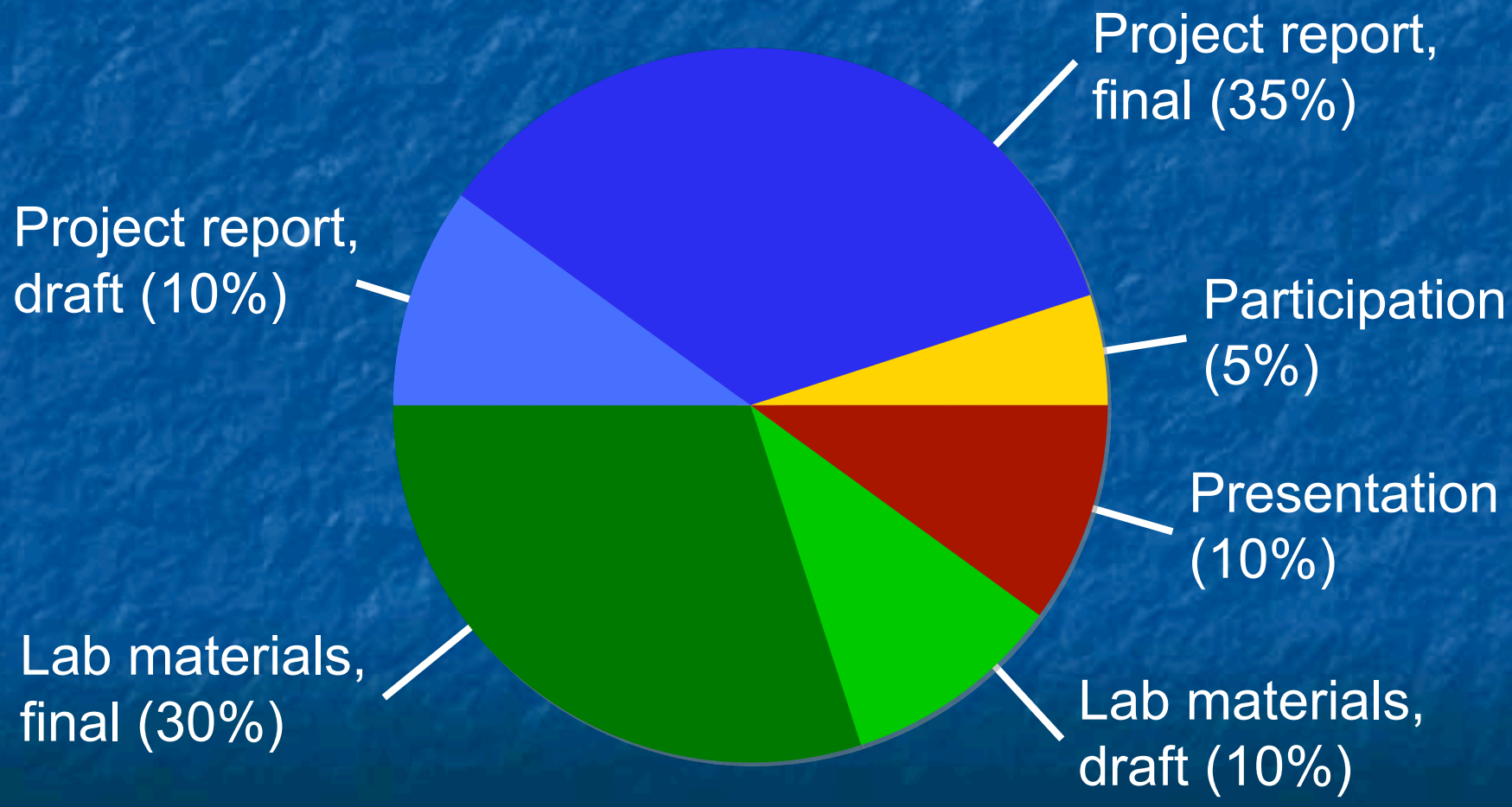


- Presentation phase: each student gives a presentation on his or her tool
- Laboratory phase: each student prepares and directs a laboratory exercise
- Comparison phase: students compare their tools

- Presentations, lab exercise, and comparison discussions each lasted 75 minutes (one of two course periods per week)

Deliverables and Grading

- Grades were based on:



Project reports

- Reports described:
  - The tool's developers
  - The problem it addresses
  - It's capabilities, strengths, and weaknesses
  - How the tool compares to others we studied
- Course staff worked closely with selected students to help them improve reports
  - Final reports were of high quality
- Reports collected into a tech. report:

  http://www.cs.virginia.edu/~pjg2e/documents/survey_mc_mbd.pdf

SCRtool Presentation

- Students read:
  - C. Heitmeyer, “Managing Complexity in Software Development with Formally Based Tools”
  - K. Heninger, “Specifying Software Requirements for Complex Systems: New Techniques and Their Applications”
- SCR team’s vision for tools
- Tabular notation and concepts
- Collected questions
  - *Audience questions focused later investigation and shaped the lab exercise*

SCRtool Laboratory

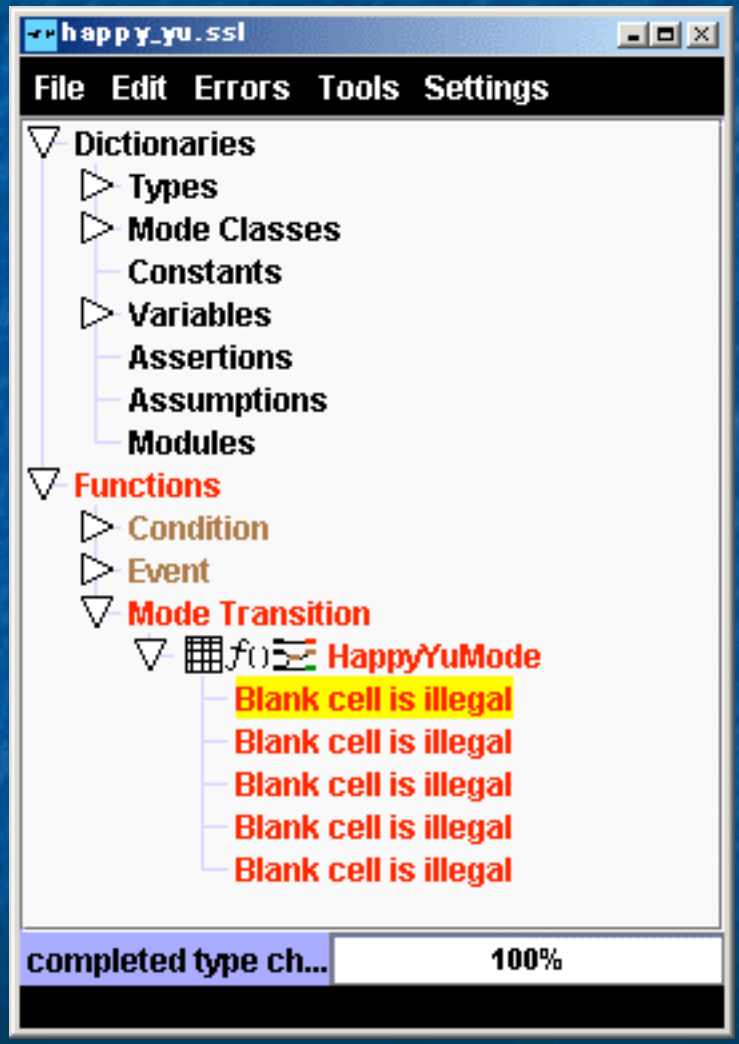
- Students completed a specification for a computer-controlled bath tub

  - *Mode transition tables* control modes for filling, heating, draining, etc.
  - *Condition tables* and *event tables* define the values of *controlled variables*

```
@T(PowerButton = Depressed)
```

SCRtool Laboratory (Cont'd)

- Students checked spec for type, coverage, and disjointness errors
  - We didn't plant errors
- Students *attempted* to prove a safety property
- We simulated the specified system



Discussion

- Model checkers and MBD tools compared and contrasted separately
- MBD discussion covered:
  - The kind of software the tool is intended for
  - The kind of developer the tool is aimed at
  - The tool's limitations
  - The guarantees made by the tool
  - The V&V activities supported by the tool
  - The tool's code generation capabilities
  - The tool's usability and scalability

Looking Back

- Students showed interest in teaching
  - Lab development offered as an alternative to a longer paper
- Comments from students positive
- Reports were overall high quality
- Tool comparisons impressive

Acknowledgement

- We thank:
  - Ralph Jeffords of the Naval Research Laboratory for his assistance in obtaining and using SCRtool
  - Escher Technologies, Esterel Technologies, iLogix, and The MathWorks for letting us use their tools
  - The students in the course, for their efforts and for trying this approach to introducing formal methods
  - Kendra Schmid, Michael Spiegel, and Benjamin Taitelbaum for their comments on SCADE, Perfect Developer, and Simulink, respectively