# BML: Specification and Verification at the Bytecode Level

Aleksy Schubert
Institute of Informatics
Warsaw University
ul. Banacha 2
02–097 Warsaw
Poland

October 14, 2008

# BML – Bytecode Modeling Language

- Bytecode specification language

# BML – Bytecode Modeling Language

- Bytecode specification language
- Proposed by: Lilian Burdy, Marieke Huisman, and Mariela Pavlova

# BML – Bytecode Modeling Language

- Bytecode specification language
- Proposed by: Lilian Burdy, Marieke Huisman, and Mariela Pavlova
- Main features:
  - similar to JML
  - based on design-by-contract principles
  - covers (JML0):
    - invariants (static & instance), history constraints, simple form of represents clauses
    - pre- and post- conditions (with exceptions), modifies clauses
    - asserts, assumes, loop invariants, decreases clauses, loop modifies clauses

# BML – Bytecode Modeling Language

- Additional features:
  - access to local variables and stack
  - compression of multiple requires-ensures, invariants, and constraints

# BML Reference Manual

- peope involved: Jacek Chrzszcz, Marieke Huisman, Aleksy Schubert,
  and Joe Kiniry, Erik Poll, Mariela Pavlova
- covers:
  - definition of the textual format
  - definition of the bytecode format
  - definition of a translation from JML to BML
- work in progress (80% ready)
- web page: `http://www-sop.inria.fr/everest/BML/`
  also available from `http://www.jmlspecs.org`

# Tools and formalisms

- BML – specification language
- JACK – Java Card verification environment
- Umbra – specification editor
- BMLLib – library to parse and store BML specifications
- JML2BML – compiler of JML to BML
- BML to BoogiePL translator

# JACK

- preliminary work on BML
- people involved: Gemplus & INRIA Everest
- features:
  - storing BML in class files
  - editing BML specifications
  - generation of proof obligations
- web page:
  `http://www-sop.inria.fr/everest/soft/Jack/jack.html`

# Umbra

- bytecode and BML specification language editor
- people involved: Jacek Chrzszcz, Tomasz Batkiewicz, Wojciech Ws, Aleksy Schubert
- features:
  - one can disassemble an existing Java source code file,
  - one can view an existing class file,
  - one can add, delete, and edit bytecode mnemonics,
  - one can add, delete, and edit JML specifications,
- web page: `http://zls.mimuw.edu.pl/~alx/umbra/`

# BMLLib

- library to manipulate the specifications,
- people involved: Jacek Chrzszcz, Tomasz Batkiewicz, and Aleksy Schubert
- features:
  - one can parse textual BML specifications
  - one can print out textual BML specifications
  - one can read BML specifications from class files
  - one can write BML specifications from class files
  - one can manipulate BML specifications programmatically
  - based on BCEL bytecode library
- web page: `http://zls.mimuw.edu.pl/~alx/umbra/`

# JML2BML

- standalone compiler of JML specifications to BML specifications
- people involved: Jdrzej Fulara, Krzysztof Jakubczyk, Aleksy Schubert
- it takes Java source code with JML annotations + compiled class file
  and returns class file with BML attributes
- web page: http://zls.mimuw.edu.pl/ alx/jml2bml/

# BML to BoogiePL

- a tool which transforms BML annotated bytecode to BoogiePL
- people involved: Ovidio Mallo, Hermann Lehner
- features:
  - reading class files with BML specifications
  - writing text files with BoogiePL result
  - based on ASM bytecode library

# CCT — embed certificates into class files

- toolset to embed certificates into class files

# BMLLib and BoogiePL

- coupling of BMLLib with BoogiePL

# Presentation

# Work in progress

- translation from BML to Coq
- translation of non-interference type system to BML

# Work in progress

- case study

# Thank you!