An Overview of Overture and how tools for VDM are bootstrapped

Rodin User and Developer Workshop

Kenneth Lausdahl
kenneth AT lausdahl.com

Miguel Ferreira
m.ferreira AT sig.nl

Aarhus School of Engineering
Software Improvement Group

Southampton — July 16 2009
Outline

The Overture Project
   Overview of the Overture Tools

Tool Set
   Components
   Eclipse integration

Bootstrapping Tools for VDM — Past and Present
   Bootstrapping
   History of VDMTools Bootstrapping — Peter Gorm Larsen [1]
   Bootstrapping Overture Tools
Introduction

- Vienna Development Method
Introduction

• Vienna Development Method

• Multi dialect (VDM-SL, VDM++ and VDM-RT).
Introduction

- Vienna Development Method

- Multi dialect (VDM-SL, VDM++ and VDM-RT).

- VDM Tools
The Overture project

Mission

Overture’s mission is twofold:

- to provide an industrial-strength tool that supports the use of precise abstract models in software development, and
- to foster an environment that allows researchers and other interested parties to experiment with modifications and extensions to the tool.

The Overture tools are being developed by volunteers, researchers and students.
Overview of the Overture Tools

Basic automatic checks and GUI
- Refactoring support
- Editor with syntax highlighting
- Syntax check
- Type check

Eclipse

Verification support
- Model checking support
- Interactive Proof support
- Automatic proof support
- Proof obligation generation

Connection to standard development environments
- Code Generators (C++, Java)
- Reverse Engineering support
- GUI generators
- UML visualisation support
- SysML, AADL visualisation support

Legend
- Not yet scheduled
- Under development
- Planned
- Completed (Stable)
Overture components

Basic automatic checks and GUI

- Refactoring support.
- Editor with syntax highlighting.
- Syntax check.
- Type check.

Connections to standard development environments

- Code Generators C++, Java.
- Reverse Engineering support.
- GUI generators.
- UML visualization support.
- SysML, AADL visualization support.
Components

Validation support

- Interpreter (with debug protocol).
- Test generation support.
- Visualization for execution traces.
- Pretty Printing with coverage.

Verification support

- Proof obligation generation.
- Model checking support.
- Interactive Proof support.
- Automatic proof support.

Connection to Rodin

The automatic proof support is currently done in HOL but the GUI presentation of that will be difficult inside Eclipse. So an alternative connection to Rodin would be interesting if possible.
Components

Validation support

- Interpreter (with debug protocol).
- Test generation support.
- Visualization for execution traces.
- Pretty Printing with coverage.

Verification support

- Proof obligation generation.
- Model checking support.
- Interactive Proof support.
- Automatic proof support.

Connection to Rodin
The automatic proof support is current done in HOL but the GUI presentation of that will be difficult inside Eclipse. So an alternative connection to Rodin would be interesting if possible.
Outline

The Overture Project
Overview of the Overture Tools

Tool Set
Components
Eclipse integration

Bootstrapping Tools for VDM — Past and Present
Bootstrapping
History of VDMTools Bootstrapping — Peter Gorm Larsen [1]
Bootstrapping Overture Tools
VDMJ

Features

• Syntax check.
• Type check.
• Interpreter (with debug protocol).
• Pretty Printing with coverage (partly).
• Test generation support.
• Proof obligation generation.
• Multi dialect: VDM-SL, VDM++, and VDM-RT
Editor

Features

- Editor with syntax highlighting.
- Syntax check.
- Type check.
- UML visualization support.
- Interpreter.
- Test generation support.
Editor Main view
Demo

- Overture Debugger.
Outline

The Overture Project
  Overview of the Overture Tools

Tool Set
  Components
  Eclipse integration

Bootstrapping Tools for VDM — Past and Present
  Bootstrapping
  History of VDMTools Bootstrapping — Peter Gorm Larsen [1]
  Bootstrapping Overture Tools
Bootstrapping

Compilers

“Bootstrapping is a term used in computer science to describe the techniques involved in writing a compiler (...) in the target programming language which it is intended to compile.”

From Wikipedia, the free encyclopedia
Bootstrapping
How is it done?

Niklaus Wirth — the first Pascal compiler
• Used a different language — first implementation was written in Fortan;
• Manually compiled the compiler — second implementation was written in Pascal and hand compiled.
Bootstrapping

How is it done?

• How to produce the first compiler for a new language?
Bootstrapping
How is it done?

• How to produce the first compiler for a new language?

Niklaus Wirth — the first Pascal compiler

• Used a different language — first implementation was written in Fortan;
Bootstrapping

How is it done?

• How to produce the first compiler for a new language?

Niklaus Wirth — the first Pascal compiler

• Used a different language — first implementation was written in Fortan;

• Manually compiled the compiler — second implementation was written in Pascal and hand compiled.
Architecture
Overview
Architecture

Conventional development
Architecture
Formally specified

- VDM-SL Module
- Corba based API
- UML Coupling
- Type Checker (SS)
- Proof Support (PS)
- Code Generators (CGs)
- External C++
- DL Modules
- Scanner Parser
- VDM-SL Module
- Word Support
- Error Tool
- Help Tool
- Automatcally generated files
- User Interface
- GUI
- Rational Rose UML
- Emacs

- User input files
- Pretty Printer
- Test Coverage
- Test Statistics
- Word LaTeX Output
- Specfication Manager (SM)
- Java/C++ Code File
- User Interface
- Parser
- Scanner
- External
- Rational Rose
- Error Tool
- Help Tool
- Word
- Word Corba based API
- ASCII or RTF
- C++ Interpreter
- Debugger (DS)
- DL Modules
- Type Checker (SS)
- Proof Support (PS)
- Code Generators (CGs)
- External C++
- DL Modules
Architecture
Generated code

The Overture Project
Tool Set
Bootstrapping Tools for VDM

ASCII or RTF
VDM-SL Module

User Interface
GUI
Error Tool
Help Tool

Word
Support

Corba based API

Corba based API

UML Coupling

Rational Rose
UML

Emacs

Pretty Printer

Test Coverage

Word LaTeX Output

Test Statistics

Automatically generated files

User Interface

User input files

Scanner Parser

DL Modules

External C++

Interpreter Debugger (DS)

Type Checker (SS)

Proof Support (PS)

Code Generators (CGs)

Java/C++ Code
File

VDM-SL Module

DL Module

User input files

ASCII or RTF

Larsen P.G.: Ten Years of Historical Development
"Bootstrapping" ...

Tool Set

Bootstrapping Tools for VDM
Bootstrapping

Test and improvement

- I/D spec
- TC spec
- PS spec
- SM spec
- CG spec

Type Checker

Interpret er Debugger

Test & Improve
Bootstrapping

Tool generation
Bootstrapping

Tool generation

The Overture Project

Tool Set

Bootstrapping Tools for VDM
Architecture
Overview

Diagram:
- AST
- Basic automatic checks and GUI
  - Refactoring support
  - Editor with syntax highlighting
  - Syntax check
  - Type check
- Eclipse
- Connection to standard development environments
  - Code Generators
    - C++, Java
  - Reverse Engineering support
  - GUI generators
  - UML visualisation support
  - SysML, AADL visualisation support
- Validation support
  - Interpreter (with debug protocol)
  - Test Generation support
  - Visualization for execution traces
  - Pretty Printing with coverage
- Verification support
  - Model checking support
  - Interactive Proof support
  - Automatic proof support
  - Proof obligation generation

Legend:
- Not yet scheduled
- Under development
- Planned
- Completed (Stable)
Architecture

Conventional development

Eclipse

Basic automatic checks and GUI

- Refactoring support
- Editor with syntax highlighting
- Syntax check
- Type check

Connection to standard development environments

- Code Generators
  - C++, Java
- Reverse Engineering support
- GUI generators
- UML visualisation support
- SysML, AADL visualisation support

Verification support

- Model checking support
- Interactive Proof support
- Automatic proof support
- Proof obligation generation

Legend

- Not yet scheduled
- Under development
- Planned
- Completed (Stable)
Architecture
Formally specified
Architecture

Generated code

Eclipse

AST

Basic automatic checks and GUI
- Refactoring support
- Editor with syntax highlighting
- Syntax check
- Type check

Interpeler (with debug protocol)

Test Generation support

Visualization for execution traces

Pretty Printing with coverage

Connection to standard development environments
- Code Generators C++, Java
- Reverse Engineering support
- GUI generators
- UML visualisation support
- SysML, AADL visualisation support

Verification support
- Model checking support
- Interactive Proof support
- Automatic proof support
- Proof obligation generation

Legend
- Not yet scheduled
- Under development
- Planned
- Completed (Stable)
Bootstrapping

AST Generation

AG
vdm++

VDMT Tools

AST Gen
Bootstrapping

AST Generation
Bootstrapping
Component Generation

HOLT vdm++

ST vdm++

UMLV vdm++

VDMTools

Overture Tools

HOLT

ST

UMLV
Thank you!

www.overturetool.org
Peter Gorm Larsen.

Ten Years of Historical Development: “Bootstrapping” VDMTools.