# VDM Tools integration with Maven

Overture Workshop

Newcastle 2009

Kenneth Lausdahl

# Agenda

- Maven integration
- Dependency resolving
- Type check
- Code generation
- Packages – Importing VDM packages
- Customizing VDMT behavior
- Example
- Exercise

# Why VDM Tools and Maven

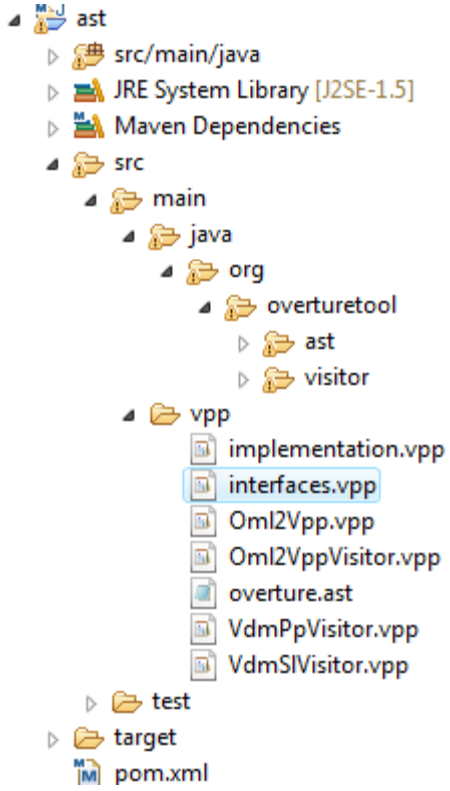
- Maven: Software tool for Java project management and build automation
- VDM Tools: VDM type check, code generation, etc.
- Easy code generation of VDM from Maven
- Make it easier to sync VDM to Java
- Structure VDM in “packages”

# The idea

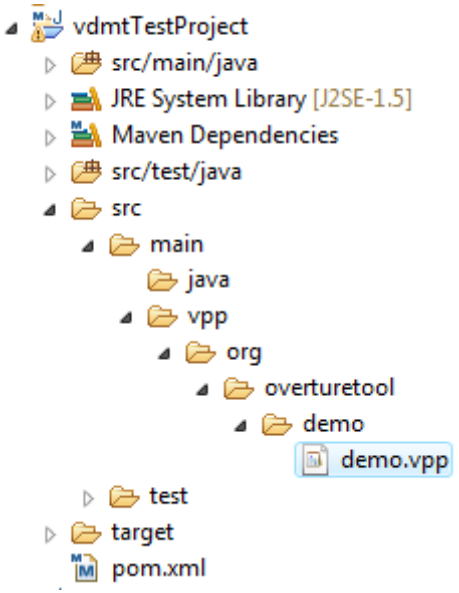

Maven
Vpp
• Type check VDM model
Maven
Vpp
• Code Generate vpp to Java
Maven
Java
• Compile java

# Dependency resolving

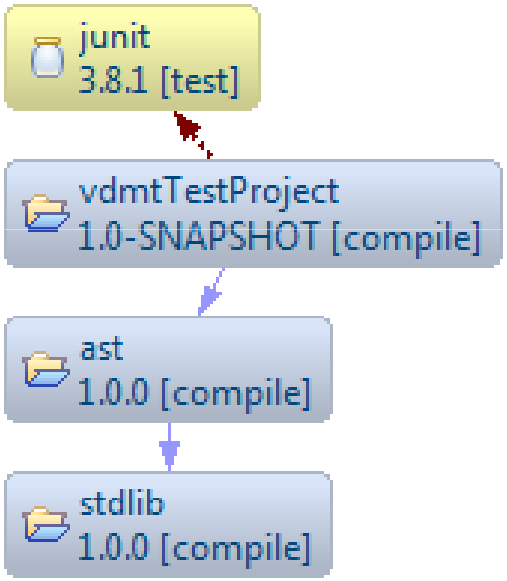
## AST project




## Demo project




## Dependency graph



# Maven Setup

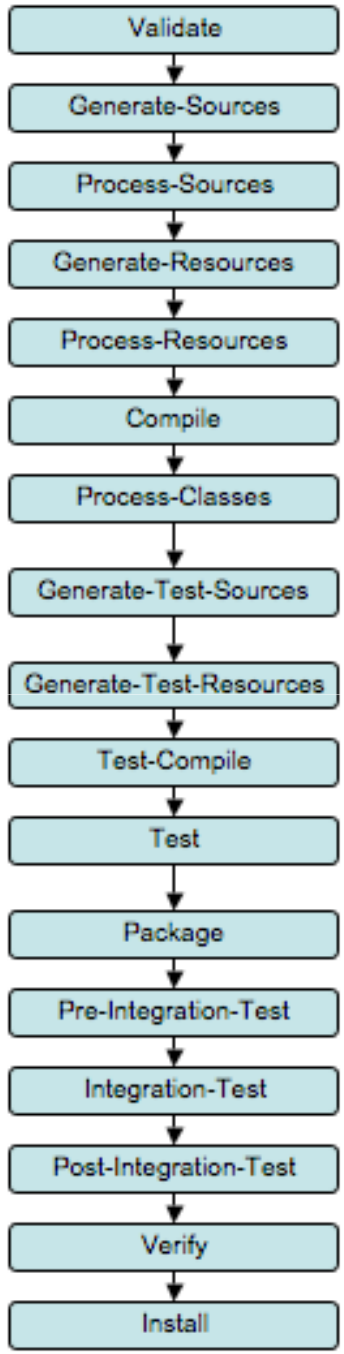
- Create a default Maven profile
  - Set the path to VDM Tools
- Settings location
  - ${user.home}/.m2/settings.xml

```
<properties>
  <user.vdmtoolscmdpath>
    c:\Program Files\The VDM++ Toolbox v8.2b\bin\vppde.exe
  </user.vdmtoolscmdpath>
</properties>
```

# Maven integration

- VDM Tools is integrated as a Mojo
- Requires dependency resolution
  - Compile →
- Enabled in the POM of a project

```
<build>
  <plugins>
    <plugin>
      <groupId>org.overturetool.tools</groupId>
      <artifactId>vdmt</artifactId>
    </plugin>
  </plugins>
</build>
```



# Type check

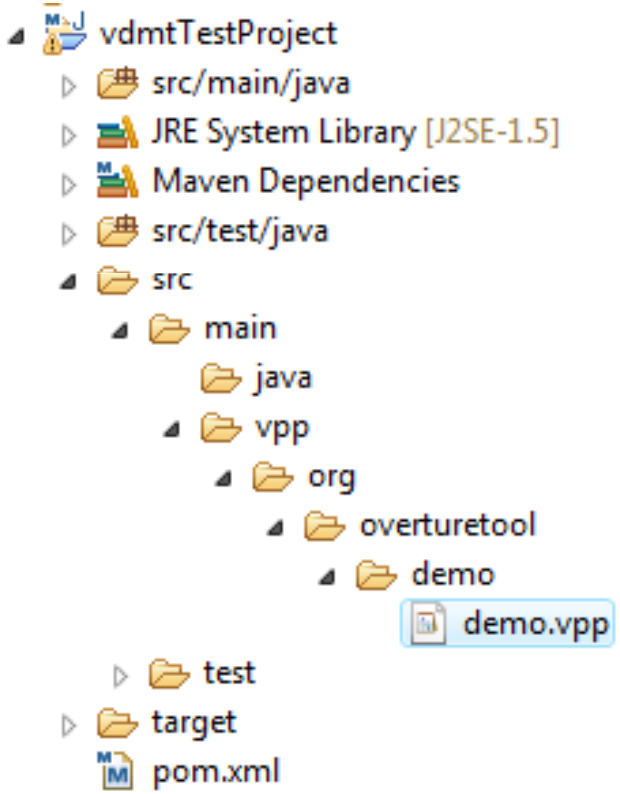
- Goal: “type”
- Terminal command:
  - “mvn org.overturetool.tools:vdmt:type”
- Description
  - Runs VDM Tools type check
  - Resolves one level dependencies
  - Vpp files is fetched from src/main/vpp in all projects

# Code generation

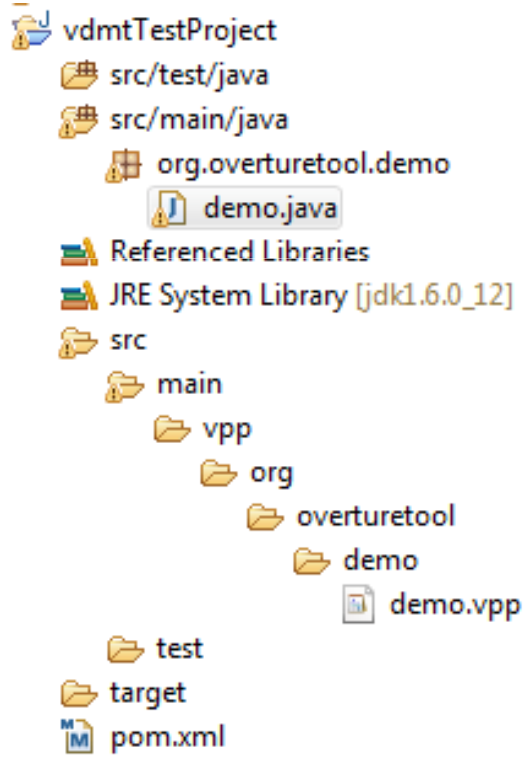
- Goal: “code”
- Terminal command:
  - “mvn org.overturetool.tools:vdmt:code”
- Description
  - Runs VDM Tools Java code generator
  - Resolves one level dependencies
  - Vpp files from current project src/main/vpp
  - Folders below src/main/vpp is used as Java packages

# Packages – Importing VDM packages

## VDM




## Java




```
import org.overturetool.ast.itf.*;
import org.overturetool.ast.imp.*;
import org.overturetool.demo.*;
```

# Customizing VDMT behavior

- Exclude vpp package
- Exclude VDM class
- Import additional Java package

```
<build>
  <plugins>
    <plugin>
      <groupId>org.overturetool.tools</groupId>
      <artifactId>vdmt</artifactId>
      <configuration>
        <excludePackages>
          <param>org.overturetool.traces.test</param>
          <param>org.overturetool.traces.VDMUnit</param>
        </excludePackages>
        <excludeClasses>
          <param>SomeClassToExclude</param>
        </excludeClasses>
        <importPackages>
          <param>org.overturetool.ast.itf</param>
          <param>org.overturetool.ast.imp</param>
        </importPackages>
      </configuration>
    </plugin>
  </plugins>
</build>
```

# Example