- Pre-requisites:
  - install cygwin (http://www.cygwin.com/)
  - install VDMTools for VDM++ or VICE (8.0.1b or later)
  - install java JDK 1.6.0

- create a abstract syntax description file, i.e. overture.ast.
  make sure the file contains the following macros:
  - " `%prefix Oml;` "
    This prefix will be added to all generated classes.
    E.g. "Test" will become "OmlTest" and "IOmlTest"

  - " `%package org.overturetool.ast;` "
    This is the top-level Java package name used for the generated code.
    The package name here is just as an example and should not be used.

  - " `%directory ".";` "
    This macro will determine to which directory the generated code is written to.
    This usually is the top-level location of the Eclipse project.

  - " `%top Specifications Expression;` "
    This macro will define which top-level abstract syntax elements contained in the Document class.

- Run the astgen tool on the ast file, from a normal windows command shell using the supplied batch file. If the ast file does not contain errors, the abstract syntax classes will be generated. This is the first step in the code generation.

- Goto the directory where the code is generated, using a bash shell in cygwin.
  Goto the "src" subdirectory. For convenience (merging the large number of files will make VDMTools significantly faster), take the following steps:
  - " `cat IOml*.tex > interfaces.tex` "
  - " `rm -rf IOml*.tex` "
  - " `cat Oml*.tex > implementation.tex` "
  - " `rm -rf Oml*.tex` "

- Open VDMTools and add the *.tex files to an empty project. The files should parse without any problems. Save this project under an arbitrary name into the "src" directory. This is important because VDMTools generates the Java code relative to the location of this file!

- Open the project settings dialog in VDMTools and goto the Java code generator settings. Reset all options, except "generate integers as longs", this one should be enabled. Also enter the name of the package, extended with ".imp". In other words "org.overturetool.ast" as used above should become "org.overturetool.ast.imp" here. Press "apply" and "ok". Verify whether or not these settings have been adopted properly. Most convenient way to do this is to close and restart VDMTools, re-open the project and check the project settings.

- Parse the four tex files, type check all classes (in the project dialog, select an arbitrary class and press CTRL-A, which will automatically select all classes, then press "type check"). All classes should be indicated as syntax and type correct. Now generate the Java code. This is step two of the generation process.

- Goto the " `src/org/overturetool/ast/imp` " directory using a bash shell
- Execute the "vdmpatch" bash shell script by typing:
    - " `./vdmpatch` "
    - " `rm vdmpatch script` "
- This completes step three of the generation process. It is now ready to be compiled in Eclipse. Find the OmlDocument class in "src/org/overturetool/ast/imp" and add " `import org.overturetool.visitor.*;` " to the import section
- Update the Eclipse project, the code should now compile without problems.
  Note that VDM.jar should be part of the external libraries included in the Eclipse project.