

VDM Tools Introduction

Aims

- ☐ To install VDM Tool
- ☐ To introduce VDM Tool, creating project files and using the interpreter
- ☐ To introduce the general format of a VDM specification
- ☐ Learn how to define basic Types, Instance Variables, Invariants
- ☐ Syntax checking, type checking and editing a model

1. Introduction

VDMTools is an industrial strength support tool for the formal modelling language VDM++. This tutorial provides an introduction to the tool and the tools that it can be connected to with advantage. Since you will need to use this tool suite substantially for your own project it is essential that you complete this material. Work through the tasks at your own pace. When you have completed the tasks you should discuss them with the teacher and get the exercise signed off.

2. Installing the tool

- Go to eLEAP.
- Download the VDM++ Tool Lite, and install it in your computer.
- You also can get the VDM++ tool from website: Overture Tool Formal Modelling in VDM, together with manuals.
- <http://overturetool.org/>
- In this course we use The VDM++ Toolbox Lite Ver8.0

3. Getting Started

1. Under windows start the GUI version of VDMTools from the start-menu all-programs/vdmttools. You should be presented with the main Toolbox window.

2. Click on Windows, then Interpreter.

3. We'll use the Interpreter later to query a model: for now, we can use it to evaluate straightforward mathematical and logical expressions. Type the following at the >> prompt, noting the answers returned after each line:

```
print 3*12
print 3 < 4
print {1,...,10}
```

Most output should be self-explanatory: notice that the interpreter can evaluate logical expressions such as $3 < 4$ as well as mathematical ones. The last expression returns a set of integers.

Other commands are available in the interpreter dialog window: the help command lists them and gives more detailed help on specific commands. Exit the toolbox.

4. Organising Projects and reading files

4.1 Specification template

A standard specification has one or more class definitions, each of which may include definitions of types, operations, instance variables, functions and values.

2. Write the following VDM++ Specification by using Notepad editor as in figure 1.

```

-----
-- Author:
-- Created:
-- Updated:
-- Description:
-----

--
-- class definition
--
class c1

--
-- instance variables
--
instance variables

--
-- Types definition section
--
types

--
-- Operations definition section
--
operations

--
-- Functions definition section
--
functions

--
-- Values definition section
--
values

end c1

```

Figure 1: VDM++ Specification

The file contains a header comment and a class definition for a class c1. Each class definition will normally have the sections shown for types, instance variables, operations, functions and values. Several classes can be defined in the same specification.

3. **Edit the header comment** by adding relevant details and then save the file under a new name.
4. Start VDMTools and begin a new project. **Add the file you just created to your project.**
5. **Syntax check and type check** the file. It's empty apart from the headings so the check should pass.

6. Click on the *Class* Tab of the *Manager* window. You should see the status of the class c1 shown with a **S** and **T** under the Syntax and Types headings.

5. Types, Instance variables, Values, Invariants

Write a formal specification about student in VDM tool. Run syntax and type checking to validate the specification written.

1. Define a class of student.

2. Define the following required data types:

2.1. Course Name

2.2 Faculty (using union type), example of faculties: fsktm, feb....

3. Define a constant value for following:

3.1 1 hour required for 1 credit hour. (Tips: 1 credit = 1 hour)

3.2 Passing GCPA Mark (Tips: 2.5)

4. Define the following attributes for a student:

4.1 Student's Name (string)

4.2 Faculty Name (IT or Engineering)

4.3 Year of study (number)

4.4 Course registered (a set of course name)

4.5 Has study Loan? (Yes or No?)

4.6 GCPA (number)

5. Define an invariant for above attributes:

5.1 Year of Study - the value of year of study should be greater than year 2016 and not more than or equal to year 2020.

5.2 GCPA – the value should not be greater than 4.0

End of Lab 1