Overture – Open-source Tools for Formal Modelling TR-2011-05
May 2011

Release Planning for Overture

by

Peter Gorm Larsen
Kenneth Lausdahl
Augusto Ribeiro

Contents

Chapter 1

Introduction

This document is intended to provide an overview of the release planning for the open source project called Overture. Efforts will be made to ensure that stable releases of the executables for Overture in a systematic fashion. It is recommended that official public releases of these executables are synchronised with each other. The planning presented here is aggressive and probably on the optimistic side but naturally this also depends upon the time the different stakeholders are able to deliver to the projects during 2010.

Official major public releases will be made yearly and will be made available through the Overture official websites. One month before the major release, a Release Candidate will be made for the purpose of testing and locate bugs. In the meantime, snapshot builds will be provided via SourceForge. In this period only bug fixes needed for the features planning in that release will be checked in. Minor official public releases will be released approximately once every three months and for each of these two weeks of release candidate periods will be used. A full release procedure will be followed for each of the public releases such that the users can expect all public releases to be stable. In between these public releases ad-hoc releases will be produced on a needs basis but it is suggested that at least a weekly build that contains the latest source code will be put in system. For such ad-hoc releases the documentation cannot be guaranteed to be updated and it will not be tested at the same level of detail as for official releases so no guarantee is provided for all ad-hoc releases. All the builds will consist of stand alone branded version for the Windows, Mac and Linux platforms.

This document start off with providing an overview of the release numbering scheme in Section 2. This is followed by Section 3 which provides the current plan for the releases of Overture executables during 2010.

Chapter 2

Release Build Numbering

The Overture project deliverables (both executables as well as associated documentation) build numbering will be made with the following format:

X.Y.Z

The **X** indicates which is the major public release version of the tool.

An increase of **Y** indicates a minor public release including significant update since last version, for example, a new feature which have been through a proper release procedure. Thus the users can expect it to be in a stable state.

An increase of **Z** indicates an ad-hoc release with minor update since the last version, for example, bug fixes. No guarantee for stability is provided for these ad-hoc releases because they have not been going through the release procedure used for public releases.

When release candidates are made available at the X and Y level there shall be a list of checks that needs to be performed by different named stakeholders before the release candidate is lifted up to before an official X or Y release. This check list shall include:

1. All examples for all dialects must be up to date with the software from the release.
2. The tutorials for all dialects must be updated to fit the newest screen dumps and the features in the new release.
3. Checks to be performed on the executable at each OS platform (Mac, Linux and Windows). The explicit list of what to check for the executables must be created.

Chapter 3

Contents of Overture releases in 2010

This section presents a plan for the contents of the 2010 releases of the Overture executable.

Release 0.3.0 (June 2010): The aim for this release is to include the following features:

1. Follow standard way to use pictures in Eclipse and document this for all developers. AugustoRibeiro. **Dependencies: None**
2. *Documentation of the Overture IDE in order to enable more developers to make updates for the IDE development to follow similar principles. This shall document the implementation of the editor, the parser, the builder, the outline and the AST access.* ***Status:*** *Ongoing* *AugustoRibeiro*, ***Dependencies: 1***
   (a) *Parser*
   (b) *Builder; How to insure TC is completed*
   (c) *How to add debug with a new interpreter and corresponding launch configuration.*
3. *Provide help (F1) inside Eclipse*, as a first go we just add pdf files to the welcome page; AugustoRibeiro
4. *provide goto definition and completion;* *AugustoRibeiro*
5. *Complete the new ASTGen utility and add all sources for it to the tools utility in the Overture SF SVN.* *NickBattle* ***Dependencies: None***
6. *Change to the AST generated from the new version of ASTGen all over the Overture sources (i.e. this will affect the development of UML, JML and POtrans components).* *NickBattle, KennethLausdahl*, ***Dependencies: 1***
7. A button for moving a test case from the combinatorial testing perspective into the debugger such that it will stop at the right place for the run-time error reported for the test case selected. KennethLausdahl see launch shortcut to see how this can be done
8. Include all definitions in the Outline view for all VDM dialects. AugustoRibeiro, **Dependencies: R020:1** Missing RT System and Threads

9. Make use of **`measure`**'s inside the the interpreter so it would be checked in the same way as invariants, pre and post conditions. NickBattle

Release 0.4.0 (September 2010): The aim for this release is to include the following features:

1. Investigate the possibility of Eclipse p2. (Adding Self-Update to an RCP Application) `http://wiki.eclipse.org/RCP_FAQ` Will do.
2. New version of the interpreter for scheduling of multiple threads in VDM++ and VDM-RT prepared for co-simulkation also able to break after a fixed amount of time. KennethLausdahl,++
3. User documentation in pdf available as a part of the Overture executable so the user can search on-line in these sources (including a VDM-10 language manual). PeterGormLarsen,++

Chapter 4

Contents of Overture releases in 2011

Release 1.0.0 (February 2011): The aim for this release was to include the following features:

1. *Quick fix incorporated in the editor view of the IDE for all VDM dialects.* ??
2. *Goto definition in the edit view of the IDE across all files in a project for all VDM dialects.* ??
3. *First version of refactoring support for all VDM dialects.* ??
4. Possibly first release of proof support feature for proving a subset of the POs generated. NickBattle
5. Support in the Proof Obligation Explorer view for new icons indicating whether the user have proved a PO or have manually inspected it and approved it. This also means that the icon with the red cross shall only be used for those where proof have failed and a less intemidating iconj shall be used for the POs that one have not yet dealt with. ??
6. *Include VDMDoc (inspired by JavaDoc) for all VDM dialects.* ??
7. *Stable version of the UML mapper enabling proper round-trip engineering between VDM and UML class diagrams and enable generation of traces from a subset of UML sequence diagrams.* *KennethLausdahl,PeterGormLarsen*
8. *First release of code generator from VDM to a programming language.* *MarcelVerhoef*
9. *Increased LaTeX support in the editor such that parts outside the* `vdm_al` *environments are highlighed as TeXClipse would do it.* ??

Chapter 5

Contents of Overture releases in 2012

Release X (?): The aim for this release is to include the following features:

1. VDM Tools
   (a) Start VDM Tools command line prompt from a Overture project. Where the prompt is a console inside Eclipse.
   (b) Create a number of buttons for VDM Tools code generation through the VDM Tools comman dline interface, with output in the console.
   (c) *Make a VDM Tools corba DBGPReader to enable a VDM Tools debugger to connect to the Overture IDE and use the basic debug facility.*
   (d) Preferences for VDM Tools
2. Ant scripts to build a product
3. Test program which can parse/type/interpret all examples and compare results with expected.
4. Extract VDM is not specified classes from Java implementation of external interpreter modules.

Release 1.0.1 (May 2011): The aim for this release is to include the following features:

1. Launching specs with errors eventually fails in IDE: ID:3206614
2. Launch dialog type checking is wrong ID: 3197397
3. PO red cross icon should change. ID: 3193289
4. XMI is not created inside the generated folder ID: 3193287

Release 1.1.0 (November 2011): The aim for this release is to include the following features:

1. Complete the new ASTGen utility and add all sources for it to the tools utility in the Overture SF SVN. NickBattle **Dependencies: None**,

2. Include VDMDoc (inspired by JavaDoc) for all VDM dialects. / Allowing multi line comments in VDM to be represented in the AST [??]
3. Make VDM ast pretty printer for math syntax

Release Future (2011/2012): The aim for this release is to include the following features:

1. Documentation of the Overture IDE in order to enable more developers to make updates for the IDE development to follow similar principles. This shall document the implementation of the editor, the parser, the builder, the outline and the AST access. **Status:** Ongoing [AugustoRibeiro], **Dependencies: 1**
   (a) Parser
   (b) Builder; How to insure TC is completed
   (c) How to add debug with a new interpreter and corresponding launch configuration.
2. Provide help (F1) inside Eclipse
3. Provide goto definition and completion; [AugustoRibeiro]
4. Change to the AST generated from the new version of ASTGen all over the Overture sources (i.e. this will affect the development of UML and POtrans components). [NickBattle, KennethLausdahl], **Dependencies: 1**
5. Quick fix incorporated in the editor view of the IDE for all VDM dialects. [??]
6. Goto definition in the edit view of the IDE across all files in a project for all VDM dialects. [??]
7. First version of refactoring support for all VDM dialects. [??]
8. Stable version of the UML mapper enabling proper round-trip engineering between VDM and UML class diagrams and enable generation of traces from a subset of UML sequence diagrams. [KennethLausdahl,PeterGormLarsen]
9. First release of code generator from VDM to a programming language. [MarcelVerhoef]
10. Increased LaTeX support in the editor such that parts outside the `vdm_al` environments are highlighed as TeXClipse would do it. [??]
11. Make Java code generator
12. Make C code generator
13. Make C++ code generator
14. Make VDM Doc parser
15. Make VDM source format
16. Tests of the plug-ins
17. Make a VDM model of Multi-core CPU distribution/scheduling.

*