Verified File Store

Latest Status

The base specification for the produced VDM++ model is the Intel Flash File System Core Reference Guide (Version 1). The verification of the model's integrity was done at three different levels:

- Testing: unit testing and prototype animation using The VDMTools;
- Model Checking: functional requirements and proof obligations are model checked in Alloy, with equivalent (hand written) models;
- Proof of Correction: automatic using the Automatic Proof System (by SanderVermolen), or manual using the PF-transform, relational calculus, proof obligation calculus.

The current models include:

- FlashFileSystemCore.(vpp/als): Global definitions of data types and auxiliary functions, which are used in multiple layers of the architecture;
- FileSystemLayerBase.(vpp/als): File System Layer definitions of data types and auxiliary functions;
- FileSystemLayerOperations.(vpp/als): Underlying functions of each API method;
    - implemented operations:
        - FS_DeleteFileDir
        - FS_OpenFileDir
- FileSystemLayerObj.vpp: Outmost class of the layer where the API methods as VDM++ operations;
- ProofObligations.als: VDMTools generated proof obligations written as Alloy asserts, that can be model checked.

There are two slices of the model:

- one for FS_DeleteFileDir,
- and another for FS_OpenFileDir (this slice also contains the specification of FS_DeleteFileDir, as it is a dependency of the FS_OpenFileDir).

The reason behind the slices is the fact that the second operation was specified on top of a previously verified specification of the first operation. It is our belief that having slices for each operation was quite helpful in the analysis and verification tasks.

All the details behind the project can be found in the MSc dissertation that resulted from the project.

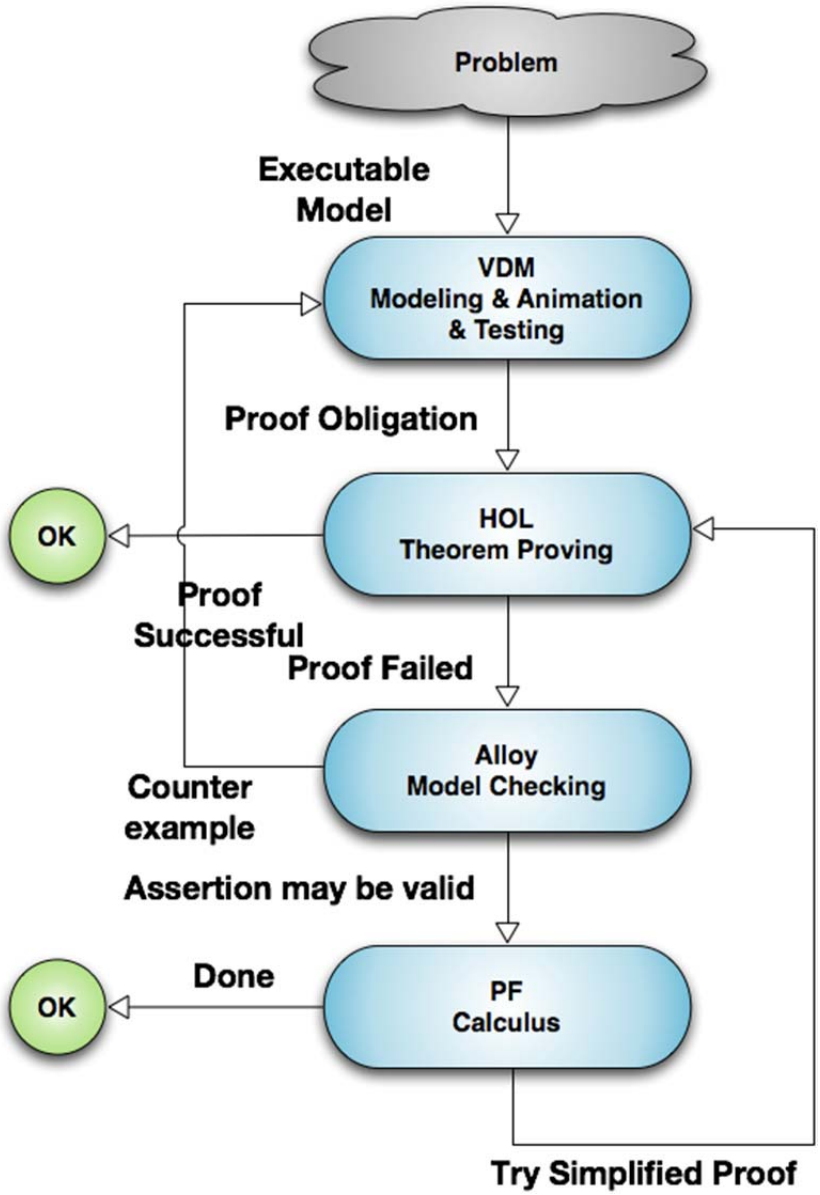
There is also a paper published in the Proceedings of the 4th VDM-Overture Workshop at FM'08, with the tile **`Verifying Intel's Flash File System Core`** (pages 54-71).

Participants

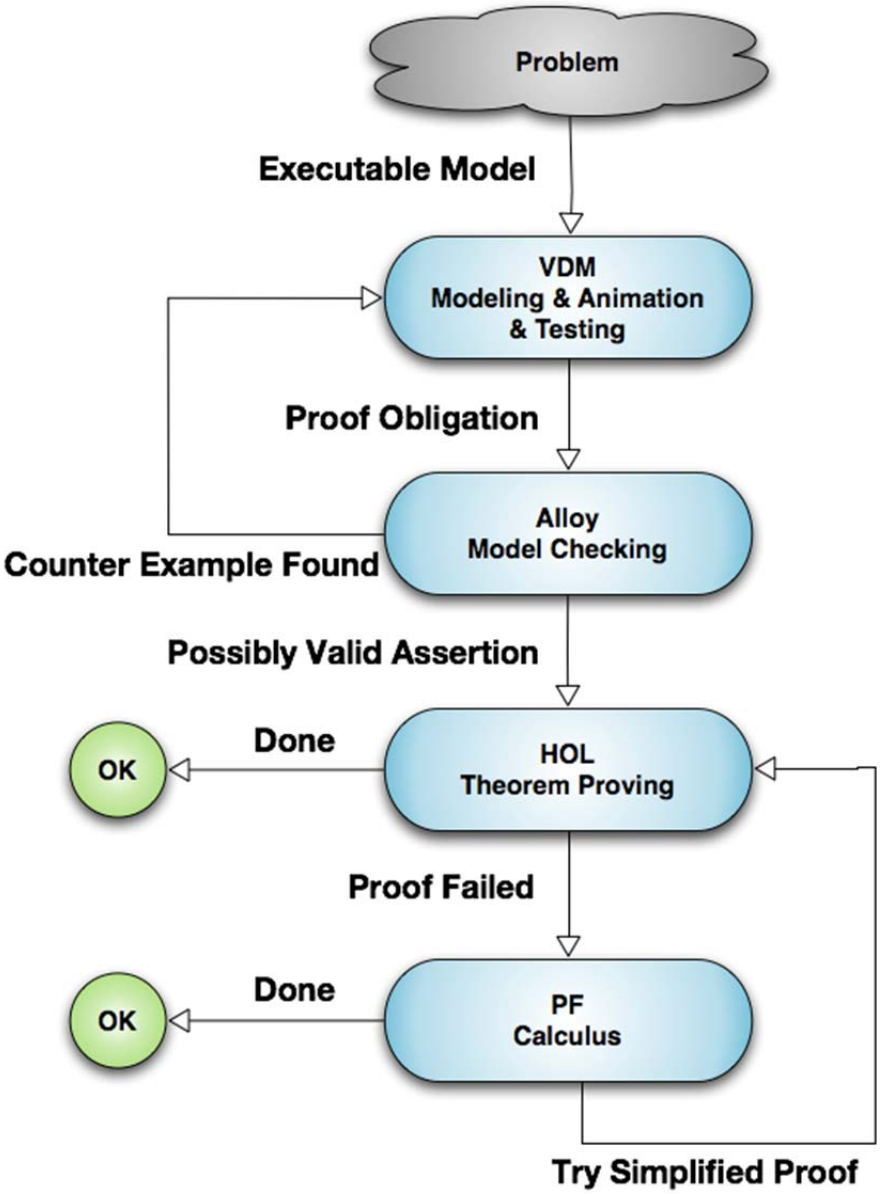
JoseOliveira, MiguelFerreira, SamuelSilva

"All-in-one" Verification Process

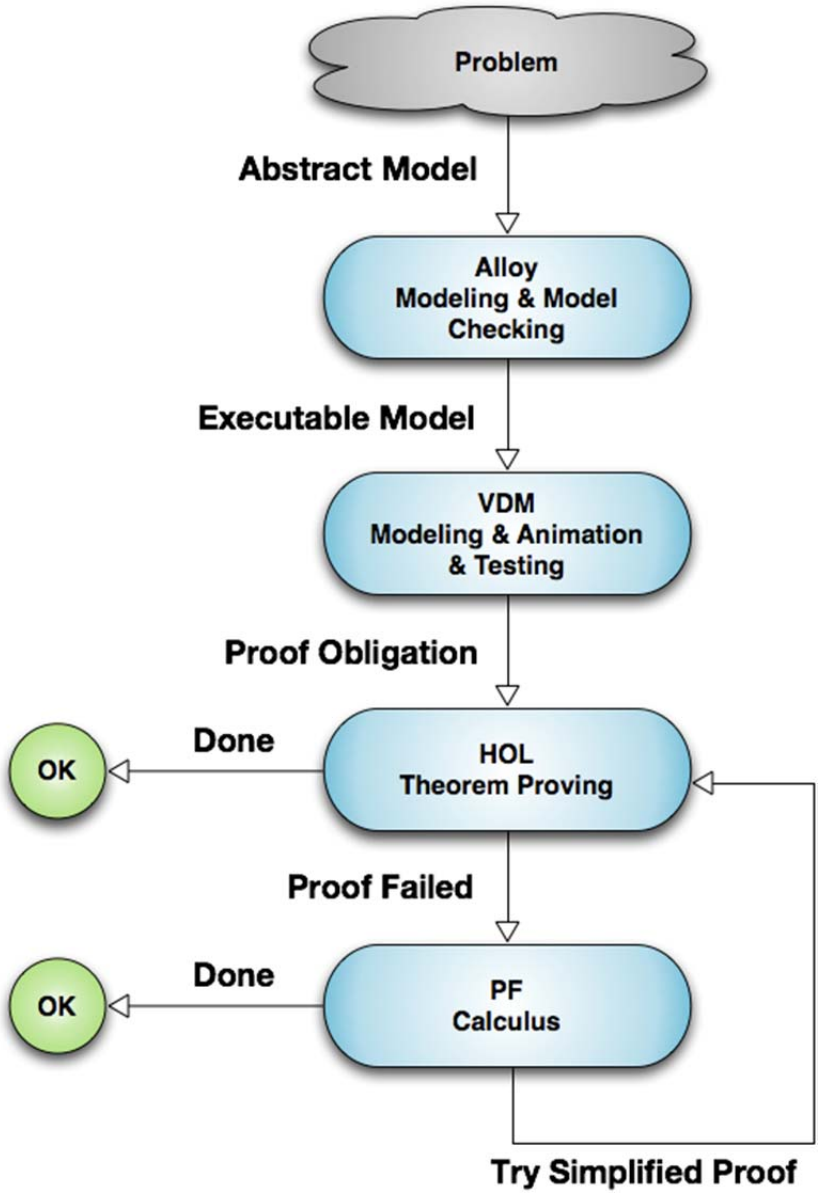
Our approach resorts to the VDMTools proof obligation generator and the VDM to HOL translator developed by SanderVermolen. The VDM to Alloy conversion is manual. In this "all-in-one" approach, modeling and testing takes place in the VDM phase. Alloy is particularly helpful in finding counter examples to proof obligations.



This verification process as evolved to an architecture where VDM++ is followed by Alloy, and only then HOL. This evolution removes the focus from automatization, and puts is in incrementally increase confidence in the specification correctness.



Furthermore, another evolution of the architecture is possible, giving priority to abstraction, where Alloy would be the first step to build an abstract model, then followed by a more concrete VDM++ model, ending with proofs in HOL.



Intel Flash File System Core Model

File System Layer Models

There has been a restructuring of all models, and for that, some of them aren't available yet. If you are looking for any thing ins specific please contact us.

VDM++

- src_vdm.tar.bz: FileSystemLayer (VDM++)
  - Implemented Operations:
    - FS_DeleteFileDir
    - FS_OpenFileDir

Alloy

- src_alloy.tar.bz: FileSystemLayer (Alloy)
    - Implemented Operations:
        - FS_DeleteFileDir
        - FS_OpenFileDir

VDM++ adapted for VdmHolTranslator and HOL translation

Model

- src_vdm2hol.tar.bz: FileSystemLayer (VDM2HOL)
    - Implemented Operations:
        - FS_DeleteFileDir
        - FS_OpenFileDir

Proof Obligations

- FS_DeleteFileDir proofs
    - The VDMTools generated 11 Proof Obligations, from which:
        - 3 aren't translatable to HOL with the VdmHolTranslator. These can be found in the **`vdm2hol/delete/excluded.pog`** file
        - 8 where automatically translated and discharged in HOL using the Automatic Proof System. These can be found in the**`vdm2hol/delete/FileSystemLayerAlg.mod.vpp.pog`** file.
    -
    - HOL model and Proof Obligations can be found in:
        - **`vdm2hol/delete/FileSystemLayerAlg.mod.vpp.pog.hol`** (for the unmodified version)
        - **`vdm2hol/delete/FileSystemLayerAlg.mod.vpp.pog.hol.mod`** (for the executable version)
- FS_OpenFileDir proofs
    - The VDMTools generated 41 Proof Obligations, from which
        - 4 aren't translatable to HOL with the VdmHolTranslator. These can be found in the **`vdm2hol/open/excluded.pog`** file
        - 34 where automatically translated to HOL by the VdmHolTranslator. These can be found in the **`vdm2hol/open/FileSystemLayerAlg.mod.vpp.pog`** file. 3 POs had to be generated from the VDM++ model with out adaptations and can be found in **`vdm2hol/open/missingpos.pog`**. This operation required some

functions to be generated from the model that as no adaptations, due to the fact that some functions had to be removed in the adaptation, and later manually translated to HOL.

- 24 of these Proof Obligations where discharged in HOL using the Overture Automated Proof Support
- for the remaining 13 Proof Obligations, HOL attempted to prove for more than 10 hours running on a dedicated node at the SEARCH cluster with no results.

- HOL model and Proof Obligations can be found in:
  - **`vdm2hol/open/FileSystemLayerAlg.mod.vpp.mod.pog.hol`** (for the unmodified version)
  - **`vdm2hol/open/FileSystemLayerAlg.mod.vpp.mod.pog.hol.mod`** (for the executable version)

Hand made proofs (point free style)

- Calculation of weakest pre-condition for preservation of referential integrity invariant on open files can be found here, on Lecture 5 (page 144).

External resources

Flash File System

- (pdf) Intel Flash File System Core Reference Guide (Version 1)

POSIX File Store (GC)

- (pdf) Morgan and Sufrin's paper on the Unix filing system. (Z)
- (pdf) POSIX file store in Z/Eves: an experiment in the verified software repository
- (ppt) Formal Modeling and Analysis of a Flash Filesystem in Alloy.
- (pdf) Verifiable POSIX file store, technical report. (VDM)
- (pdf) Open Nand Flash Interface, technical report. (VDM)

Tools

- Overture - Open-source Tools for Formal Modeling
- VDMTools
- Alloy
- HOL

- Extractors: these are two parsers, a VDM++ class and a bash script, developed during the project to speed up preparation of VDM++ models for the Automatic Proof System. Both parsers were developed in an ad hoc fashion, and are not based on a sound grammar of the input files language (as they should!). For more information about these tools see MSc dissertation and the README file inside the archive.

Conferences and Workshops

- VS-EXPERIMENTS Workshop, VSTTE'09, 9 October 2008
- ABZ'08 conference, London, 16-18 September 2008
- VDM-Overture 4th Workshop, FM'08, Turku, 26 May 2008
- Pilot Projects for the Grand Challenge in Verified Software, FM'08, Turku, 27 May 2008
- Workshop on the Verifiable File Store Mini-Challenge, BCS offices, London, 18 Dec 2007

Other

- The Verified Software Initiative: A Manifesto
- Verified File-System
- Verified Software Repository
- OLVER (Open Linux VERification)
- Linux