# The HOL-UNITY verification system

Flemming Andersen Ulla Binau Karsten Nyblad
Kim Dam Petersen Jimmi S. Pettersson

Tele Danmark Research, Lyngsø Allé 2, DK-2970 Hørsholm

**Abstract.** The HOL-UNITY verification system consists of a collection of tools for specifying and verifying UNITY programs and their properties. All the tools interface the theorem prover HOL for proving the properties of UNITY programs. In this way HOL-UNITY supports mechanised proving of correctness for parallel programs.

## Description

A goal of the HOL-UNITY system is the development of aids for the production of reliable software within the telecommunications industry. The present system consists of several tools:

1. A theorem prover, which is actually a mechanisation of the UNITY theory [1, 5] with some extensions in the Cambridge Higher Order Logic theorem prover HOL [2, 3]. The extensions are mainly support for restricted properties similar to Beverly Sanders' subscripted properties [4]. For automatically proving basic UNITY properties (safety and ensures) specialised tactics are implemented in the HOL system.
2. A compiler, which translates UNITY programs, properties, and proofs into HOL representation. The compiler recognises an extended UNITY language, which includes sequential programming constructs, program modules, restricted properties, Chandy and Misra style natural deduction proofs, HOL-style proofs, etc.
3. A graphical tool for developing annotated proof lattices for leadsto properties and translating them into natural deduction style UNITY proofs. Annotated proof lattices are similar to Lamport and Owicki proof lattices but specialised to the UNITY logic, and supplied with annotations which guides the theorem prover in proving the basic properties of the lattices.
4. A compiler for translating verified UNITY programs into executable CC++ (Compositional C++) programs.

The HOL-UNITY system has been used to verify a number of smaller examples such as: mutual exclusion, readers and writers, a two-way arbiter, a lift-control, and a sliding window protocol. All these examples have been exercises towards its application on real problems.

Especially the lift-control example has demonstrated how these tools can be used together for developing a program and property specification, verifying

that the program satisfies the required properties, and finally generating an executable program. The lift-control example was developed by first specifying the required properties, and then without taking verification perspectives into consideration the program was specified. This strategy was used to achieve a more realistic scenario for the verification.

The next steps towards realistic applications is the currently on-going task of specifying and verifying an ATM-protocol using the HOL-UNITY tools and the development of tactics for reasoning about service and feature interaction.

**Environment:** Preferably DecStation 5000/200, but also Sun SPARC, or Sun 3/60 running Ultrix and Standard ML of New Jersey, version 0.93 with 64MB RAM and 150 MB swap

**Contact:** For further information please email: fa@tdr.dk, or look into WWW URL: http:/www/TDR-Home.html

## References


1. K. Mani Chandy and Jayadev Misra. *Parallel Program Design: A Foundation.* Addison–Wesley, 1988.
2. M. J. C. Gordon and T. F. Melham. *Introduction to HOL. A theorem proving environment for higher order logic.* Cambridge University Press, Computer Laboratory, 1993.
3. K. Slind. HOL90 Users Manual. Technical report, 1992.
4. Beverly A. Sanders. Eliminating the Substitution Axiom from UNITY Logic. *Formal Aspects of Computing*, 3(2):189–205, April-June 1991.
5. Flemming Andersen. *A Theorem Prover for UNITY in Higher Order Logic.* PhD thesis, Technical University of Denmark, 1992. Also published as TFL RT 1992-3, Tele Danmark Research, 1992.
6. F. Andersen, K.D. Petersen, and J.S. Pettersson. *Program Verification using HOL-UNITY.* In *Higher Order Logic Theorem Proving and Its Applications, 6th International Workshop, HUG'93*, LNCS 780, pages 1–15, 1993.
7. F. Andersen, K.D. Petersen, and J.S. Pettersson. *A Graphical Tool for Proving Progress.* In *Higher Order Logic Theorem Proving and Its Applications, 7th International Workshop, HUG'94*, LNCS 859, pages 17–32, 1994.
8. F. Andersen, K.D. Petersen, and J.S. Pettersson. *Verification of Software.* Teleteknik, Vol. 1–2, 1993, English Edition. Pages 66–75.
9. U. Binau. *Correct Concurrent Programs: A UNITY design method for Compositional C++ programs.* PhD thesis, Technical University of Denmark, 1994.
10. U. Binau. *Mechanical Verification of a CC++ Mutual Exclusion Program in HOL-UNITY.* Technical report in preparation, Tele Danmark Research, 1994.
11. K.D. Petersen and J.S. Pettersson. *Proving Protocols Correct – Proving Safety and Progress Properties of the Sliding Window Protocol using HOL-UNITY.* Research Report TFL RR 1993-3, Tele Danmark Research, December 1993.