Designing Reliable, High-Performance Networks with the Nuprl Logical Programming Environment

— Position Statement —

Christoph Kreitz
Department of Computer Science, Cornell University
Ithaca, NY 14853, U.S.A.

Formal methods tools have greatly influenced our ability to increase the reliability of software and hardware systems. Extended type checkers, model checkers and theorem provers have been used to detect subtle errors in prototype code and to clarify critical concepts in system design. Automated theorem proving now has the potential to support a formal development of reliable systems at the same pace as designs that are not formally assisted, provided it is engaged at the earliest stages of design and implementation.

An engagement of deductive methods at this stage depends on a formal language that is able to naturally express the ideas underlying the software systems, a knowledge base of formalized facts about systems concepts that a design team can use in its discussions, and a theorem prover capable of integrating a variety of different proof techniques while providing assurance for the correctness of the joint result.

The NUPRL **L**ogical **P**rogramming **E**nvironment (Constable *et al.* 1986) is a framework for the development of formalized mathematical knowledge as well as for the synthesis, verification, and optimization of software. It provides an expressive formal logic (Martin-Löf 1984; Constable 1998) and a theorem proving environment (Constable *et al.* 1986; Allen *et al.* 2000; NuPRL) that supports interactive and tactic-based proof development, extraction of programs from proofs, program evaluation, language extensions through a definition mechanism, and an extendable library of verified algorithmic knowledge.

Our goal is to demonstrate that the NUPRL LPE is capable of supporting the formal design and implementation of large-scale, high-performance network systems. We have already used the NUPRL LPE in the verification of protocols for the ENSEMBLE group communication toolkit (Kreitz, Hayden, & Hickey 1998; Hickey, Lynch, & van Renesse 1999), in verifiably correct optimizations of ENSEMBLE protocol stacks (Kreitz 1999; Liu *et al.* 1999), and in the formal design and implementation of new adaptive network protocols (Liu *et al.* 2001; Bickford *et al.* 2001a; 2001b). Currently we are working on providing formal support for the development of large distributed embedded systems.

Our experience shows that logical methods that have proven effective in program synthesis, verification, and optimization can be made to scale up to large software systems by employing several layers of formal abstraction and compositional reasoning techniques, by building large libraries of verified algorithmic knowledge, and by continuously expanding the logical foundations and automated proof capabilities of the reasoning environment.

Copyright © 2002, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

References

Allen, S.; Constable, R.; Eaton, R.; Kreitz, C.; and Lorigo, L. 2000. The Nuprl open logical environment. In McAllester, D., ed., *17th Conference on Automated Deduction*, *LNCS* 1831, 170–176. Springer Verlag.

Bickford, M.; Kreitz, C.; van Renesse, R.; and Constable, R. 2001a. An experiment in formal design using meta-properties. In Lala, J.; Maughan, D.; McCollum, C.; and Witten, B., eds., *DARPA Information Survivability Conference and Exposition II (DISCEX 2001)*, volume II, 100–107. IEEE Computer Society Press.

Bickford, M.; Kreitz, C.; van Renesse, R.; and Liu, X. 2001b. Proving hybrid protocols correct. In Boulton, R., and Jackson, P., eds., *14th International Conference on Theorem Proving in Higher Order Logics*, *LNCS* 2152, 105–120. Springer Verlag.

Constable, R. L.; Allen, S. F.; Bromley, H. M.; Cleaveland, W. R.; Cremer, J. F.; Harper, R. W.; Howe, D. J.; Knoblock, T. B.; Mendler, N. P.; Panangaden, P.; Sasaki, J. T.; and Smith, S. F. 1986. *Implementing Mathematics with the NuPRL proof development system.* Prentice Hall.

Constable, R. L. 1998. Types in logic, mathematics, and programming. In Buss, S. R., ed., *Handbook of Proof Theory*. Elsevier Science Publishers B.V. chapter X, 684–786.

Hickey, J.; Lynch, N.; and van Renesse, R. 1999. Specifications and proofs for Ensemble layers. In Cleaveland, R., ed., *5th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, *LNCS* 1579, 119–133. Springer Verlag.

Kreitz, C.; Hayden, M.; and Hickey, J. 1998. A proof environment for the development of group communication systems. In Kirchner, C., and Kirchner, H., eds., *15th Conference on Automated Deduction*, *LNAI* 1421, 317–331. Springer Verlag.

Kreitz, C. 1999. Automated fast-track reconfiguration of group communication systems. In Cleaveland, R., ed., *5th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, number *LNCS* 1579, 104–118. Springer Verlag.

Liu, X.; Kreitz, C.; van Renesse, R.; Hickey, J.; Hayden, M.; Birman, K.; and Constable, R. 1999. Building reliable, high-performance communication systems from components. In *17th ACM Symposium on Operating Systems Principles (SOSP'99)*, volume 34 of *Operating Systems Review*, 80–92.

Liu, X.; van Renesse, R.; Bickford, M.; Kreitz, C.; and Constable, R. 2001. Protocol switching: Exploiting meta-properties. In Rodrigues, L., and Raynal, M., eds., *International Workshop on Applied Reliable Group Communication (WARGC 2001)*, 37–42. IEEE Computer Society Press.

Martin-Löf, P. 1984. *Intuitionistic Type Theory*, volume 1 of *Studies in Proof Theory Lecture Notes*. Napoli: Bibliopolis.

Nuprl home page.
http://www.cs.cornell.edu/Info/Projects/NuPrl.