AN INTRODUCTION TO THE CONSTRUCTION AND VERIFICATION OF ALPHARD PROGRAMS

Wm. A. Wulf, Carnegie-Mellon University
Ralph L. London, USC Information Sciences Institute
Mary Shaw, Carnegie-Mellon University

ABSTRACT

The programming language Alphard is designed to provide support for both the methodologies of "well-structured" programming and the techniques of formal program verification. Language constructs allow a programmer to isolate an *abstraction*, specifying its behavior publicly while localizing knowledge about its implementation. The verification of such an abstraction consists of showing that its implementation behaves in accordance with its public specifications; the abstraction can then be used with confidence in constructing other programs, and the verification of that use employs only the public specifications.

This paper introduces Alphard by developing and verifying a data structure definition and a program that uses it. It shows how each language construct contributes to the development of the abstraction and discusses the way the language design and the verification methodology were tailored to each other. It serves not only as an introduction to Alphard, but also as an example of the symbiosis between verification and methodology in language design. The strategy of program structuring, illustrated for Alphard, is also applicable to most of the "data abstraction" mechanisms now appearing.