Developing Tools for Formal Methods

P.Kefalas, G.Eleftherakis and A.Sotiriadou

City Liberal Studies, Affiliated College of the University of Sheffield,
Computer Science Department, 13 Tsimiski Str.,
54624 Thessaloniki, Greece
{kefalas,eleftherakis,sotiriadou}@city.academic.gr

Abstract. Formal methods are based on rigorous mathematical notations, which aim to describe systems in the early stages of software development. Such formal descriptions are useful to precisely specify a system's data and/or control, and as a consequence to verify whether certain properties are true as well as to test whether the final product meets the initial description. All of these stages are crucial in the development process and researchers involved with formal methods claim that "correct" software can only be achieved through the use of formal methods. However, practitioners are skeptical against formal methods, arguing that, apart from the lack of training, the lack of tools to support formal development is the major drawback. In this paper, we present a framework for developing such tools for formal methods by describing the requirements and the steps that are necessary to meet this objective. The outline process will facilitate tool developers to identify the necessary steps to be taken, i.e. to define a practical core specification language, to compile such language to some form of executable code and use this code as means to aid further tool support for verification, model checking, testing etc. We record our experience on this proposed process by giving an example of a set of tools developed around a specific formal method, which we present as a case study.

1. Introduction

The extensive use of computer systems, as integrated parts of almost any engineered product has brought up two important issues, safety and reliability. The need for more robust, reliable and safe software and hardware and the fact that errors in various different stages of the production of a computer system can be responsible for the creation of non-reliable systems, has started to seriously concern the community 35 years ago [1]. Traditional computer system development methods proved to be inadequate to develop reliable and safe software. In the early days of this software crisis, it was a general belief that formal methods will become mainstream software development techniques in the future.

Over the last years, it is widely admitted that use of formal methods in software engineering is essential [2], while there are several cases proving the applicability of formal methods in industrial applications [3] showing very good results. However, many practitioners are still reluctant to adopt formal methods. The main reasons are: (a) other software engineering techniques successfully increased system quality by improving processes and methodologies, and by using friendly tools, thus allowing