Definition and Basic Properties of the Deva Meta-Calculus

Matthias Weber
Technische Universität Berlin, Berlin, Germany.

Keywords: Formalization of development methods; Logical frameworks

Abstract. This article presents the definition and some basic properties of the Deva meta-calculus, a generic logical framework whose design was driven by the needs arising from the instantiation to software development methods. As a result, Deva contains structures that do not occur in comparable logical frameworks. There now exist a number of case studies about the formalization of software development methods in Deva. In this article, a structured definition of Deva is presented and basic parts of its language theory, viz Church-Rosser, closure, and strong normalization, are summarized.

Introduction

The current situation with respect to the application of formal software development methods is characterized by a partial utilization of the benefits offered by the formal approach to software development: more and more systems are successfully specified in formal notations, however formal developments down to efficient implementations and inclusive of all necessary proofs are still very rare.

The formal specification of software systems has been relatively successful because it helps yield precise knowledge and understanding of the problem to be solved: information which will effect significant benefits in the subsequent development phase. The formal development of software systems, in the above sense, has been much less successful because with growing size of examples, the transitions and proofs quickly become too complex to be effectively understood.

1 Major parts of the work reported here were carried out while the author was under contract at the German National Research Centre (GMD) in Karlsruhe and at the University of Karlsruhe

Correspondence and offprint requests to: Matthias Weber, Technische Universität Berlin, FB20, FR 5-6, 1000 Berlin 10, Germany, email: we@cs.tu-berlin.de
and managed, regardless of the degree of automatic support. The enormous amount of necessary proofs, which are mostly of technical nature, is, in the author's view, not inherent to the software development process as such, but is due to a lack of well designed and generally accepted methods and notation, a lack of a library of generally accepted standard theories for software development, and a lack of tools which support all this.

On the other hand, theoretical research has produced remarkable progress towards the development and investigation of powerful generic proof frameworks, or meta-calculi, based on typed λ-calculus, that offer a natural and universal formal development machinery and can be instantiated by various logical and mathematical theories ([Bru80], [CH85], [HHP86]). Most of these frameworks have been designed and used within a theoretical environment, to study issues of formalization and proof.

This article reports on the definition and theoretical investigation of a generic development framework designed to express completely formal developments. This framework, named Deva meta-calculus, can be seen as a contribution towards linking work performed on proof frameworks with work on formal development methods. The connection consists of viewing developments as proofs and development methods as theories. As a result, development methods become completely formalized objects. The purpose of Deva is to experiment with complete formalizations of currently known methods and with the resulting method support systems. This experimentation can be a quite rewarding process, e.g. by pointing out informal parts both in the method itself and in its use. Similarly, it is interesting to experiment with libraries of basic theories for software developments based on the method.

While our investigations about requirements and design of a development calculus dates back to the work of Michel Sintzoff [Sin80], the Deva meta-calculus presented in this article was developed mainly between 1987 and 1989 in the context of the ESPRIT-project ToolUse [SWGC89]. The objective of the ToolUse project was to study a broad spectrum of development methodologies (e.g. Jacksons System Design, the Vienna Development Method, or Burstall-Darlington's fold/unfold method for program transformation) and to design a method-driven support environment. Deva was then intended to serve as a notational framework to help clarify the understanding of such methods. The language was developed by a collaborative effort of three different subgroups of the project: a group headed by Michel Sintzoff at the University of Louvain (UCL), a group headed by Renée Jacquart at the French research centre for technology (CERT) in Toulouse, and a group headed by Stefan Jähnichen at the German research centre for computer science (GMD) in Karlsruhe. There now exist a number of case studies about the formalization of development methods in Deva ([BS91], [Gab91], [Laf90], [Web90], [Web91b]). Recently, an introduction to Deva and two extensive case studies have been compiled into a book [WSL93]. The language theory for a kernel calculus of Deva was developed by Philippe de Groote and is described in [Gro90]. This article deals with the full calculus as investigated in [Web91a].

The essential technical requirements driving the design of Deva have been the following:

- It must be possible to instantiate Deva to an interesting range of methods. Thus, it should offer a generic logical system since many methods are based on completely different logical systems.