EDEN: An Engine for Definitive Notations
Design, Implementation and Evaluation

by
Edward Yun Wai Yung

A thesis submitted for the award of
a Master's degree by research in Computer Science

Department of Computer Science
University of Warwick
Coventry CV4 7AL
U.K.

In 2 volumes:

Volume I: MSc. Thesis
Volume II: Appendices

18/9/89

Volume I

MSc. Thesis

EDEN: An Engine for Definitive Notations
Design, Implementation and Evaluation

by

Edward Yun Wai Yung

Supervisor: Dr. W. M. Beynon

Department of Computer Science
University of Warwick
Coventry CV4 7AL
U.K.

A thesis submitted for the award of
a Master's degree by research in Computer Science

Abstract

This thesis concerns the ideas and experiences of programming in EDEN which is an experimental language designed to be a general-purpose definitive (definition-based) language with some imperative features. EDEN supports a very simple form of definition—an uni-directional relationship among variables. In this thesis, we present the fundamental philosophy of the design of this language and the methods of implementing such an interpreter. We investigate the advantages and draw-backs of the current definitive approach by using EDEN as a prototyping language; we describe how a subset of DoNaLD—a definitive notation for line drawing—is implemented using the EDEN interpreter as the underlying computing engine. We compare the approach of definitive programming with those of other programming paradigms, including functional programming and constraint-based programming. The possibility of using definitive languages for software specification is investigated. We also explore the potential for developing definitive programming into a concurrent programming paradigm. Finally, we propose some possible ways of improving the current design of EDEN through introducing other programming concepts, such as object-oriented programming techniques.

Keywords: CAD, Constraint, Declarative languages, Definition, Definitive (definition-based) programming, Dependency maintainer, Formula, Functional/applicative languages, Graphics, Parallel computation, Procedural/imperative languages, Software specification.

18/9/89

Table of Contents

Volume I: MSc. Thesis

Table of Figures

Table of Listings

Acknowledgements

I wish to express my gratitude to my friends and colleagues who helped me and encouraged me during these years. In particular, many thanks to Dr. Steve Russ and Mike Slade for their useful comments.

Above all, I am most grateful to my supervisor, Dr. Meurig Beynon, for the continuous help, encouragement, and constructive criticisms he has given me during the work.

Finally, I wish to thank my family in Hong Kong for the love and moral support they showed during my stay in U.K.