Definitive Programming:
A Paradigm for Exploratory Programming

by
Simon Yun Pui Yung

A thesis submitted for the degree of
Doctor of Philosophy in Computer Science

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

October 1992

Definitive Programming:
A Paradigm for Exploratory Programming

Simon Yun Pui Yung

Supervisor: Dr W M Beynon

Department of Computer Science
University of Warwick
Coventry, U.K.

A thesis submitted for the degree of Doctor of Philosophy

Abstract

Exploratory software development is a method that applies to the development of programs whose requirement is initially unclear. In such a context, it is only through prototyping and experimenting on the prototypes that the requirement can be fully developed. A good exploratory software development method must have a short development cycle. This thesis describes our attempt to fulfil this demand. We address this issue in the programming language level. A novel programming paradigm – *definitive (definition-based) programming* – is developed.

In definitive programming, a state is represented by a set of definitions (a *definitive script*) and a state transition is represented by a redefinition. By means of a definition, a variable is defined either by an explicit value or by a formula in terms of other variables. Unless this variable is redefined, the relationship between the variables within the definition persists.

To apply this state representation principle, we have developed some *definitive notations* in which the underlying algebras used in formulating definitions are domain-specific. We have also developed an agent-oriented specification language by which we can model state transitions over definitive scripts. The modelling principles of definitive programming rest on a solid foundation in observation and experiment that is essential for exploratory software development.

This thesis describes how we may combine definitive notations and the agent-oriented programming concept to produce software tools that are useful in exploratory software development. In this way, definitive programming can be considered as a paradigm for exploratory programming.

Keywords: definitive programming, programming languages, rapid prototyping, state-transition model, modelling and simulation, software development, agents, human-computer interaction.

Table of Contents

Table of Listings

Table of Figures

Acknowledgements

I wish to express my gratitude to my friends and colleagues who helped me and encouraged me during these years. In particular, I thank Dr. Steve Russ in Warwick and Dr. Rick Thomas in Leicester University for their useful comments. I must thank my brother, Edward Yung for his love, care and practical help while he was with me in the University.

I am most grateful to my supervisor, Dr. Meurig Beynon, for the continuous help, encouragement and constructive criticism he has given me during the work. Especially, I thank him for allowing me to have a year off for developing my personality and communication skills during the course.

I am so much thankful to the Coventry Chinese Christian Church who supported me during my financial hardship so that I may concentrate on my study. And above all, I have to thank God for providing me both the necessary intelligence for the work.