Macro processors and their use in implementing software

P.J. Brown

November 1968 (revised April 1971)

Originally submitted for the degree of Doctor of Philosophy in the University of Cambridge

Table of Contents

Preface

This is the complete dissertation, apart from the two Appendices (see below). The following points should be noted:

a. The whole dissertation has been converted to Texinfo format. This has necessitated changing the Appendices from “I and II” to “A and B”.

b. Part I was published in *Annual Review in Automatic Programming*, Volume 6, Pergamon Press, 1968.

c. Each of the three Parts is self-contained and can be read independently of the other two. A knowledge of ML/I as summarised in the paper “The ML/I Macro Processor” [*Comm. ACM* **10**, 10 (Oct. 1967), 618–623] is assumed in Part II. The above paper described the main principles of ML/I and may be looked upon as a precis of Appendix B.

d. The contents of Part III are also summarised in a published paper, which is entitled “Using a macro processor to aid software implementation”, [*Computer J.* **12**, 4 (Nov. 1969) 327–331].

e. There have been two different versions of the Bibliography. These have been combined, and numbered references in the text have been adjusted accordingly.

f. The Appendices are separate documents. Both of them have been updated at least twice, and the originals are no longer easily available.

Overall Introduction

Summary

The main research for this dissertation has consisted of designing and implementing a macro processor, which I have called *ML/I*, and in using it to generate versions of itself for several computers. A description of ML/I has been published [*Comm. ACM* **10**, 10 (Oct. 1967), 618–623].

This dissertation consists of three Parts and two large Appendices.

Part I is a survey of existing macro processors, including ML/I, with an evaluation of some of their uses, their achievements and their failings. The survey introduces the four main application areas for macro processors, which are considered to be language extension, language translation, text generation, and systematic editing and searching. It then considers the design factors that make macro processors fundamentally different from one another. These design factors are: relationship with base language, syntax, text evaluation, macro-time facilities and implementation methods. The bulk of the survey (Chapter 2, Chapter 3, Chapter 4, Chapter 5 and Chapter 6) is devoted to considering these design factors in turn. The last Chapter, Chapter 7, reviews the four application areas in the light of what has been said in the preceding Chapters.

Part II is a short critique of ML/I. It is a general discussion rather than a very detailed one. A complete description of ML/I is given as Appendix A.

Part III introduces the idea of a *DLIMP*. This is a means whereby machine-independent software can be implemented by describing it in a special-purpose language and then using a macro processor to map this language into any desired object language, in particular into the assembly language of any desired machine. The implementation of ML/I itself by means of a DLIMP is described and results for several implementations are presented. Full details of how to implement ML/I by this method are given in Appendix B. The object of Part III is to show that a DLIMP can be a very good method of software implementation and that ML/I is an especially suitable vehicle for performing a DLIMP.

Acknowledgements

I would like to thank Professor M.V. Wilkes, who was my supervisor during the first and third years of my research, and Professor D.W. Barron who was my supervisor during the second year. I would like to thank John Nicholls and other members of the staff of IBM (UK) Laboratories for useful discussions and for arranging the typing of the draft of Part I. I would also like to thank the people who helped with the various implementations of ML/I. Full acknowledgements to these people are given in Chapter 12. Finally I would like to thank the Science Research Council for the financial support of my work.

Originality

No survey of macro processors nor any analysis of the basic principles in the design of macro processors has ever been published before. Part I of this dissertation is intended to fill this gap. Chapter 3 and Chapter 4, in particular, contain new insights into the design of macro processors.

ML/I, like any new piece of software, contains many facilities that have been in use before. However, its most central characteristic is original. The degree to which ML/I is original and the extent to which it has taken ideas from other macro processors is discussed at more length in Chapter 8.

The idea of a DLIMP as presented in Part III is not new, though no analysis of its virtues as a general method for software implementation has been published previously. The implementing of ML/I by means of a DLIMP is new in the following ways:

a. It is a much more elaborate operation than the only previously published account of a DLIMP.
b. New techniques such as statement prefixes and constant-defining macros have been introduced.
c. The same descriptive language has been mapped into both a high-level language and an assembly language.
d. Detailed results of a DLIMP have never been published before but the results quoted in this dissertation are thought to set a high standard of efficiency.

I hereby declare that this dissertation is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other University. I further state that no part of my dissertation has already or is being currently submitted for any such degree, diploma or other qualification.

University Mathematical Laboratory
Cambridge, England.

March 1968.