Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

292

The Munich Project CIP

Volume II: The Program Transformation System CIP-S

By the CIP System Group:
F.L. Bauer, H. Ehler, A. Horsch, B. Möller,
H. Partsch, O. Paukner, and P. Pepper

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo

Authors

F. L. Bauer
H. Ehler
B. Möller
Institut für Informatik der Technischen Universität München
Postfach 202420, 8000 München 2, Federal Republic of Germany

A. Horsch
O. Paukner
Klinikum rechts der Isar, Rechenzentrum Block A
Ismaninger Straße 22, 8000 München 80
Federal Republic of Germany

H. Partsch
Department of Informatics VI, Catholic University of Nijmegen
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands

P. Pepper
Fachbereich 20 Informatik, Technische Universität Berlin
Franklinstraße 28/29, 1000 Berlin 10 (West)

CR Subject Classification (1987): D.1.0, D.2.1–2, D.2.4, D.2.6–7, D.2.9–10, F.3.1, I.1.3, I.2.2–3, K.6.1, K.6.3–4

ISBN 3-540-18779-0 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-18779-0 Springer-Verlag New York Berlin Heidelberg

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210

PREFACE

This book is the second of two volumes that present the main results having emerged from the project CIP - Computer-Aided, Intuition-Guided Programming - at the Technical University of Munich. The central theme of this project is program development by transformation, a methodology which is felt to become more and more important.

Whereas Volume I contains the description and formal specification of a wide spectrum language CIP-L particularly tailored to the needs of transformational programming, the present Volume II contains the description, formal specification, and transformational development of a system CIP-S, that is to assist a programmer in this methodology.

This work originated from two rather different motivations: On the one hand, it is to be seen as an attempt to gain methodical experience with non-toy, medium-size software projects and, in this way, to demonstrate the feasibility of the CIP approach as a software engineering discipline. On the other hand, the system is intended to incorporate recent ideas as well as experience with our own prototype system and other people's systems. Thus, in the very end, it is to constitute the basis for a practicable software development tool usable by other people either in gaining experience themselves or in producing software.

Part I deals with general issues such as "Why to use an implemented system to assist in transformational programming?" and "What are the interesting aspects with respect to transformation systems?". It also gives a brief summary of the running CIP prototype transformation system and an informal overview of the system to be dealt with in all subsequent parts. A short account of the global requirements and their implications for the organization of the system project is given and some aspects of an appropriate user environment conclude this part.

In Part II a calculus of program transformations (including induction) is presented as a theoretical basis for the entire transformation system project.

Part III starts with a more detailed and in particular more user-oriented informal collection of technical requirements for the transformation system. In its main part a formal, algebraic specification (including all design decisions) for the language-independent core of such a system can be found, whereas language-dependent aspects and issues of an appropriate user environment are deferred to Part VI. Part III closes with a kind of validation of the formal specification and a summary of experiences made in writing the formal specification.

Part IV takes Part III as a basis and demonstrates for selected system functions how running programs can be derived from the respective specifications by means of transformations. The main criterion for selection was the probable interest of the derivations. Therefore obvious or less interesting developments deliberately have been left out. As to the derivations selected, although actually done in very small steps by using the prototype system, particular emphasis has been laid on expressing the essential lines of thought rather than particular concrete rules. However, these rationales of design also have been supplemented with enough technical information such that an interested reader should be able to redo the detailed developments himself. The essential purpose of giving these selected developments in Part IV is to demonstrate that they can be done with an implemented transformation system. Many further developments of functions specified in Part III have been carried out with the prototype. They can be found in full detail in [Ehler et al. 87], out of which Part IV has actually been selected. Like Part III, Part IV closes with a summary of the experiences gained when doing the actual developments.

Part V is a collection of transformation rules used in Part IV. According to the philosophy of the language CIP-L used for specification and development these rules are differentiated into rules for the scheme language, rules for particular data types, and rules connected to particular computation structures.

Part VI is intended to give the main hints on how to extend the system core as specified in Part III to a running system exemplified with a sublanguage of CIP-L. In particular this part contains some more information on the language-dependent types (that have been left "open" in Part III), on converters between external and internal program representations, about the way of treating context conditions, semantic relations, and meta-predicates.

The report also contains an index of sorts, objects, and operations introduced, where the given page number refers to the defining occurrence in the specification. Cross-references within one part are given by section numbers only; references to other parts are made by prefixing the respective section numbers with the (roman) part numbers.

We would like to express our thanks to the Deutsche Forschungsgemeinschaft who has sponsored this research within the Sonderforschungsbereich 49 "Programmiertechnik" for ten years. Also, we gratefully acknowledge valuable criticism by the members of IFIP Working Group 2.1, notably by R. Bird, P. King, C.H. Lindsey, L.G.L.T. Meertens, S.A. Schuman, and, above all, M. Sintzoff. Moreover, we would like to thank H. Remus and D.E. Shough from the Santa Teresa Laboratory of IBM for their continuing support. Last, but by no means least, we gratefully acknowledge many helpful remarks by our (present or former) colleagues R. Berghammer, C. Delgado Kloos, F. Erhard, U. Hill-Samelson, R. Obermeier, H.-O. Riethmayer, G. Schmidt, R. Steinbrüggen, M. Wirsing, and H. Wössner as well as the speedy and competent assistance by M. Glashauser in doing the developments on the prototype system and in preparing the typescript.

Munich, October 1987

The CIP System Group

TABLE OF CONTENTS