

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

86

Abstract Software Specifications

1979 Copenhagen Winter School
January 22 – February 2, 1979
Proceedings

Edited by D. Bjørner



Springer-Verlag
Berlin Heidelberg New York 1980

Editorial Board

W. Brauer P. Brinch Hansen D. Gries C. Moler G. Seegmüller
J. Stoer N. Wirth

Editor

Dines Bjørner
Department of Computer Science, Technical University of Denmark
DK-2800 Lyngby/Denmark

AMS Subject Classifications (1979): 68-02, 68A05, 68A30
CR Subject Classifications (1974): 5.21, 5.23, 4.20

ISBN 3-540-10007-5 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-10007-5 Springer-Verlag New York Heidelberg Berlin

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks. Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to the publisher, the amount of the fee to be determined by agreement with the publisher.

© by Springer-Verlag Berlin Heidelberg 1980
Printed in Germany

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

DEDICATION

This volume is dedicated to:

PROFESSOR, DR. TECHN. HEINZ ZEMANEK

The dedication occurs in connection with Professor Zemaneks 60th Anniversary.

The dedication is motivated by Professor Zemaneks inspired founding, and inspiring leadership of the IBM Vienna Laboratory.

The applied scientific work of the Vienna Laboratory could not have taken place had it not been for Zemaneks continued attention. Examples of this work are: the VDL (Vienna Definition Language) based, operational semantics definition of PL/I, of the mid-to-late 1960s; and the therefrom distinct VDM/META-IV (Vienna Development Method / Meta Language) based denotational semantics definition of a PL/I subset, of the early-to-mid 1970s. The editor of this volume, the co-director of the 1979 Copenhagen Winter School on 'Abstract Software Specifications', and many others, derived from - shorter or longer stays with - the Vienna Laboratory of those periods, long and lasting impressions and inspirations.

The editor of this volume is grateful to be able to open and close this volume with papers by Professor Zemanek. Both were presented at the above-mentioned Winter School. The editor is also grateful to the Springer-Verlag for the opportunity and kind permission to present this dedication.

ACKNOWLEDGEMENTS

The present collection of sixteen papers relate to the subject of ABSTRACT SOFTWARE SPECIFICATIONS. This was the theme of a two week Winter School held at the Technical University of Denmark, January 22 - February 2, 1979. The 'School' also featured a number of workshop sessions, and drew some 120 participants from 23, mostly European, countries. Most of the papers derive directly from series of lectures or workshop seminar presentations given.

The event was sponsored by, and held under the auspices of:

- (1) the Commission of the European Communities, in particular the EEC/CREST Subcommittee on Education in Informatics, and
- (2) the Danish Research Councils for the Natural- and the Technical Sciences, colloquially known as SNF & STVF.

Most valuable direct, as well as indirect (participant scholarship) financial support was also received from:

(3) the edp group ('dif data') of the Danish Engineering Society, and from the following, national IBM Companies:

- (4) IBM Denmark,
- (5) IBM Finland,
- (6) IBM Norway,
- (7) IBM Sweden, and
- (8) The Netherlands IBM (Holland).

The School Directors, D.Bjørner & C.B.Jones, wishes hereby to extend their most sincere thanks to the EEC/CREST, the SNF+STVF, the DIF-DATA, and especially to the above IBM national companies whose generous understanding and prompt contributions was most appreciated.

The Director of the Winter School finally thanks the administration & technical services of the Technical University of Denmark for having provided such outstanding facilities and services.

FOREWORD

The subject of abstractly specifying software - before embarking upon costly realizations - is presently being firmly established. Not just in University Computer Science Curricula, but also as an engineering practice in small and large corporations.

This volume records a number of software abstraction and design methods, their mathematical foundations and use. Common to the methods dealt with in this volume is their reliance on mathematical foundations. This also sets these methods apart from most other recorded means of software specification.

Most papers, with the exceptions being papers number 5, 10 and 15, cover rather exactly lectures and shorter seminars given at the 1979 Copenhagen Winter School on Abstract Software Specifications. Professors Liskov and Plotkin lectured on 'Abstractions in CLU', respectively 'Towards a Mathematical Theory of Concurrently Executing Programs'. Plotkin's paper is an elaboration of only a part of his lectures at the Winter School.

The Winter School was held at the Technical University of Denmark, in the period: January 22 - February 2, 1979.

BROAD CLASSIFICATION

Two main streams of definitional styles are identified: the constructive methods based on the Scott-Strachey approach to Mathematical Semantics, and the Algebraic Semantics methods. The papers by Stoy, Jones, Bjørner, Park and Plotkin belong to the former school; with the papers by Zilles, Dahl, Burstall & Goguen, Dömölki and Liskov, the latter indirectly, belonging to the latter school.

Two hitherto separate areas of application are identified: specification of essentially deterministic, sequential, respectively non-deterministic, parallel-process oriented systems. This is admittedly a rather gross delineation. The papers by Dahl, Lauer et.al., Park and Plotkin address the latter issues, while remaining papers primarily, if not exclusively, stay within a simpler, non-power domain of discourse (!).

PAPER OVERVIEWS

The opening paper by ZEMANEK: "ABSTRACT ARCHITECTURE" relates the task of the computer and software systems architects to that of 'conventional' (i.e. building) architects; investigates the nature of design; of systems and their components; analyzes the notions of in- & formality; etc.. It is a thought provoking paper which, in very relevant terms, is a contribution towards the philosophy- and the theory of science of computer science and software engineering. Awareness of the many points brought up by Zemanek should lead to better suited, more appropriately proportioned systems serving satisfied users.

STOYS paper on the "FOUNDATIONS OF DENOTATIONAL SEMANTICS" opens the part on constructive definition methods. It provides an elementary introduction to the mathematical theory underlying such constructive definition methods, & thus, in particular, the papers by Jones and Bjørner. The papers of Park and Plotkin are more advanced treatises, extending & applying these foundations in the search for answers to and characterizations of crucial notions in non-deterministic & parallel programs. Stoy's paper also discusses techniques for reasoning about denotational semantics definitions, and for modelling GOTOs via the technique of so-called continuations. A section of Jones' paper ('Escape Mechanisms') and Bjørner's 2nd paper (numbered:5) deals with another way of modelling GOTOs - the so-called exit-mechanism. Bjørner's paper (no.5) also combines exit & continuation modelling techniques.

JONES's paper is an introduction to techniques, and a notation, for "MODELING PROGRAMMING LANGUAGE CONCEPTS". The paper unfolds the notation and techniques required, by covering concepts in a stepwise, first orthogonal, subsequently combined fashion.

BJØRNER's paper on "FORMALIZATION OF DATA BASE MODELS" provides an alternative introduction, but now to techniques for modelling Data Base concepts. Not that there is any significant difference! But the audience might be different. Jones' paper emphasizes understanding the modelling techniques and motivates (desired properties of) the notational constructs. Bjørner's paper emphasizes the application of these techniques to other than the 'classical' area (of programming languages). Jones' paper, in a sense, assumes some familiarity of the exemplified (source) language constructs; while Bjørner's paper can be read as an alternative intro=

duction to Data Base concepts for persons not familiar with these, and as an introduction to modelling techniques for Data Base professionals.

BJØRNERs paper on "EXPERIMENTS IN BLOCK-STRUCTURED GOTO LANGUAGE MODELING: EXITS VERSUS CONTINUATIONS" is a mere exercise in expressing GOTO semantics. It starts with the so-called exit-based modelling techniques motivated in Jones' paper. It then exemplifies 'corresponding' continuation-based models; and finally 'merges' these styles!

Summarizing the papers by STOY, JONES and BJØRNER, we can say that with this volume two prominent variations on the theme of expressing mathematical semantics has been brought together: The VDM (Vienna Development Method) and the Oxford Styles of Denotational Semantics. Stoy's paper clearly points out some differences, but is otherwise a contribution to a unified understanding of their foundations.

The mostly theoretical paper by ZILLES, "AN INTRODUCTION TO DATA ALGEBRA", opens the part on Algebraic Semantics. It provides an advanced level introduction to the mathematics underlying algebraic presentations of abstract data types. It is a long expected paper from one of the first researchers of this most fascinating and booming area.

DAHLs paper, "TIME SEQUENCES AS A TOOL FOR DESCRIBING PROGRAM BEHAVIOUR", explores the (time) sequence concept of e.g. programming languages, using techniques akin to those treated by Zilles. The aim is to provide a "tool kit" for speaking about 'operators, functions and predicates on sequences', aiding practicing programmers in program specification, mechanization and proofs. Use of the established tools are then demonstrated by applications to specification & proofs concerning semaphores, mutual exclusion, deadlock, and the classical readers/writers problem.

"THE SEMANTICS OF CLEAR, A SPECIFICATION LANGUAGE", by BURSTALL & GOGUEN, defines a basically algebraic specification language, CLEAR. CLEAR permits the configurated, bottom-up, as well as the hierarchical, top-down, construction of abstract models, put together, respectively derived from models of constituent, respectively overall concepts. CLEAR is here defined using 'a blend of denotational semantics with categorical ideas'.

DÖMÖLKIs paper, "AN EXAMPLE OF HIERARCHICAL PROGRAM SPECIFICATION", applies ideas of CLEAR to the development of a program: specification, realization and correctness proofs.

"MODULAR PROGRAM CONSTRUCTION USING ABSTRACTIONS", by LISKOV, 'presents a programming method in which modular decomposition is based on recognition of useful abstractions'. The paper is structured around a very instructive specification & implementation example.

LUCAS's paper "ON THE STRUCTURE OF APPLICATION PROGRAMS" is concerned with the 'parameterization of programs with respect to factual information'. It reports on 'software techniques which can be expected to facilitate programming and maintenance of commercial applications'. The paper also 'sheds light on the rôle of formalization, and the rôles and proper place of abstract data types'.

The joint paper by GERSTMANN & OLLONGREN has been included in this volume since it attempts to analyze, from one viewpoint, basic notions of the VDL-, the VDM- and the Algebraic Schools of Software Specifications.

The editor would here like to take the opportunity to warn the reader of a possible source of confusion. VDL is not VDM! The former stands for the notation language used for the operational semantics definitions of the 1960s. The latter acronym for a whole development method starting with denotational semantics definitions. VDL reads: Vienna Definition Language. VDM reads: Vienna Development Method. The notational system, or the semantics definition meta-language of VDM has been referred to by the acronym: META-IV.

'COSY' is a language for the "DESIGN & ANALYSIS OF HIGHLY PARALLEL & DISTRIBUTED SYSTEMS". It is based on Petri-net like concepts, and is derived from regular expressions. In their paper, LAUER, SHIELDS & BEST, introduces the 'COSY' notation (Lauer), gives the net semantics of 'COSY' (Best), and presents firing sequence- and adequacy properties of 'COSY' (Shields).

The last two technical papers, by Park and Plotkin, focuses on very specific, mathematical problems of dealing with parallelism and non-determinism.

PARK applies the relational semantics variant of denotational semantics in his paper "ON THE SEMANTICS OF FAIR PARALLELISM". In it, he analyzes fairness, or finite delay properties of processes, and 'unbounded, but finite', and 'potentially infinite' attributes of parallel systems specifications.

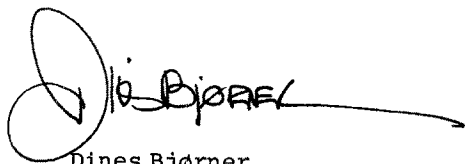
Abstract (Machine, Abstract Operational) Non-deterministic State Transformation Semantics explications of "DIJKSTRAS PREDICATE TRANSFORMERS" have been provided by e.g. de Roever, de Bakker, Wand and Bäck. PLOTKINS paper, whose title finishes with: "& SMYTHS POWER DOMAINS", 'regards this by showing homo- and isomorphisms from the state transformation view to the predicate transformer view'.

As a prerequisite for more fully enjoying the closing "BANQUET TALK" paper by ZEMANEK, the reader should be informed that (i) the above-mentioned winter school lectures took place in auditorium 81 of the Technical University of Denmark, and (ii) that "only" 50, out of a total of more than 130 participants ('students', workshopppers and lecturers) went to a mid-course Royal Danish Ballet evening which, in addition to classical, retrospective, Danish Bournonville ballet, also featured the more abstract 'Serenade' ballet by Balanchine.

CLOSING REMARKS

The 1979 Copenhagen Winter School on Abstract Software Specifications had a final panel session of some 60 minutes duration. The editor regrets being unable, at this time, to include an edited transcript of that most clarifying and concluding event. He does hope, however, some day, to be able to furnish such a written record; and invites readers to inquire.

At the banquet, where Professors Dahl and Naur entertained the more than 110 diners with several most enjoyable pieces of Bach (Naur: flute, Dahl: a somewhat out of tune piano), participants had contributed to a Winter School Song book. The editor also regrets to be likewise unable to provide this as an appendix to this volume. Since he particularly enjoyed the personal creations of Richard L. Wexelblat (of UNIVAC, Penn., USA), readers might likewise persuade the editor to provide a copy of that songbook.



Dines Bjørner

Holte, Easter 1980