M. Bidoit C. Choppy (Eds.)

Recent Trends in Data Type Specification

8th Workshop on Specification of Abstract Data Types joint with the 3rd COMPASS Workshop
Dourdan, France, August 26-30, 1991
Selected Papers

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editors

Gerhard Goos
Universität Karlsruhe
Postfach 6980
Vincenz-Priessnitz-Straße 1
W-7500 Karlsruhe, FRG

Juris Hartmanis
Cornell University
Department of Computer Science
4130 Upson Hall
Ithaca, NY 14853, USA

Volume Editors

Michel Bidoit
LIENS, C.N.R.S. U.R.A. 1327 and Ecole Normale Supérieure
45, Rue d'Ulm, F-75230 Paris Cedex, France

Christine Choppy
L.R.I., C.N.R.S. U.R.A. 410 and University of Paris-Sud
Bat. 490, F-91405 Orsay Cedex, France

CR Subject Classification (1991): D.2.1-2, D.2.4, D.2.10-m, D.3.1-3, F.3.1-2

ISBN 3-540-56379-2 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-56379-2 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Typesetting: Camera ready by author/editor
45/3140-543210 - Printed on acid-free paper

Preface

The algebraic specification of abstract data types has been a flourishing research topic in computer science since 1974. The main goal of this work is to evolve theoretical foundations and a methodology to support the design and formal development of reliable software. The Eighth Workshop on Specification of Abstract Data Types was held jointly with the Third COMPASS Workshop, 26-30 August 1991, in the town of Dourdan, about 50 kilometers from Paris, and was organized by Michel Bidoit and Christine Choppy. The main topics covered by this joint workshop were:

- specification languages and program development
- algebraic specification of concurrency
- theorem proving
- object-oriented specifications
- order-sorted algebras
- abstract implementation and behavioural semantics.

The general feeling was that this joint workshop was extremely successful and that the contributions were of a high scientific level. Part of this success is due to E. Astesiano, H.-D. Ehrich, P.D. Mosses, and F. Orejas who accepted to prepare in depth surveys. The program committee (Michel Bidoit, Christine Choppy, Hartmut Ehrig, Bernd Krieg-Brückner, Fernando Orejas, Horst Reichel, and Don Sannella) selected a number of talks which represented the most interesting ideas and reflected the main trends in current research, and asked their presenters to contribute papers. The present volume contains the final version of the selected papers, together with the invited surveys. All of them underwent a careful refereeing process, and we are grateful to the following people who agreed to referee the papers:

E. Astesiano, J. Bergstra, G. Bernot, M. Bidoit, M. Broy, C. Choppy, I. Classen, M. Cerioli, H.-D. Ehrich, H. Ehrig, G. Ferrari, W. Fey, J.F. Fiadeiro, M.-C. Gaudel, M. Gogolla, J. Goguen, M. Grosse-Rhode, I. Guessarian, M. Kindler, M. Korff, H.-J. Kreowski, B. Krieg-Brückner, J. Loeckx, P.D. Mosses, F. Nickl, F. Orejas, P. Pepper, A. Poigné, G. Reggio, H. Reichel, D. Sannella, H. Schmidt, O. Schoett, G. Scollo, T. Stroup, A. Tarlecki, N. Van Diepen, U. Waldmann, E.G. Wagner, J. Zeyer.

Thanks to Evelyne Jorion and Hélène Outin for secretarial support, and to Gilles Bernot, Pierre Dauchy, Clément Roques, and Frédéric Voisin for helping with the workshop organization. The workshop was sponsored by the Université Paris-Sud, the Ecole Normale Supérieure and the C.N.R.S., and received financial support from the ESPRIT Basic Research Working Group COMPASS, the C.N.R.S., and the C.N.R.S. GDR de Programmation.

We would like to dedicate this volume to Stéphane Kaplan.

Orsay, August 1992 — Michel Bidoit and Christine Choppy

IN MEMORIAM

Stéphane Kaplan (1961–1991)

Stéphane Kaplan was a student of the Ecole Normale Supérieure in Mathematics; he became "agrégé" in Mathematics in 1981. He received his "Thèse de 3ème Cycle" diploma and his "Thèse d'Etat" diploma in 1982 and 1987 respectively, both in Computer Science from the University of Paris-Sud, Orsay, France.

He was appointed by the C.N.R.S. (Centre National de la Recherche Scientifique) since 1983 as a "Chargé de Recherches" (full-time researcher) at the L.R.I. (Laboratoire de Recherche en Informatique) at the University of Paris-Sud. He spent a year in 1985–1986 at the Weizmann Institute of Science, Rehovot, Israel, as a visiting scientist; he came back to Israel, on leave from C.N.R.S., from 1987 to 1990, while he taught as a senior lecturer in Computer Science at the University of Bar-Ilan (Ramat-Gan), and partly at the Hebrew University of Jerusalem.

He mainly contributed to advances in rewriting (in particular in conditional rewriting), and in the specification of concurrent systems (through the theory of "process specifications" that he developed).

He was bright, dynamic, open, intuitive and fast. In addition to his qualities as a research scientist, he was a very nice guy, with a good (and light) sense of humour, a love for music, and deep human insight.

Those of us who knew him will surely miss him a lot.

Table of Contents

Algebraic Specification of Concurrency*

Egidio Astesiano and Gianna Reggio

DISI
Dipartimento di Informatica e Scienze dell'Informazione
Università di Genova – Italy
{ astes , reggio } @ cisi.unige.it

Introduction

Let us first summarize what algebraic specification is about, following [66]. Algebraic specification methods provide techniques for data abstraction and the structured specification, validation and analysis of data structures.

Classically, a (concrete) data structure is modelled by a many-sorted algebra (possibly term-generated); various categories of many-sorted algebras can be considered, like total, partial, order-sorted, with predicates and so on. An isomorphism class of data structures is called an *abstract data type* (shortly *adt*) and an *algebraic specification* is a description of one or more abstract data types. There are various approaches for identifying classes of abstract data types associated with an algebraic specification, which constitute its semantics: initial, terminal, observational; a semantics is *loose* when it identifies a class (usually infinite) of adt's.

Since data structures can be very complex (a flight reservation system, e.g.), *structuring* and *parameterization* mechanisms are fundamental for building large specifications.

Together with a rigorous description of data structures, algebraic specifications support stepwise refinement from abstract specifications to more concrete descriptions (in the end, programs) of systems by means of the notion of *implementation* and techniques for proving *correctness* of implementations. In this respect formal *proof systems* associated with algebraic specifications play a fundamental role. Finally *specification languages* provide a linguistic support to algebraic specifications.

The purpose of the algebraic specification of concurrent systems is to specify structures where some data represent processes or states of processes, i.e. objects about which *it is* possible to speak of dynamic evolution and interaction with other processes; more generally we can consider as the subject of algebraic specification of concurrency those structures able to describe entities which may be active participants of events. Such data structures will be called simply "concurrent systems", where "concurrent" conveys different meanings, from "occurring together" to "compete for the same resources" and to "cooperate for achieving the same aim".

The aim of this paper is twofold: to analyse the aims and the nature of the algebraic specifications of concurrency and to give, as examples, a short overview of some (not all) relevant work.

* This work has been supported by COMPASS-Esprit-BRA-W.G. the project MURST 40% "Metodi e specifiche per la concorrenza" and by "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo" of C.N.R. (Italy).

In Sect. 1 we introduce some basic concepts and terminology about processes: the various models around which the specification models are built and the fundamental issues of (observational) semantics, formal description and specification; moreover we give some illustrative examples of specification problems to be used later for making more concrete some general considerations and for assessing different methods.

In Sect. 2 we try to qualify the field: indicating three different fundamental motivations/viewpoints (and distinguishing between methods and instances); then outlining the issues to deal with; finally illustrating by two significant examples/approaches how this field stimulates innovations and improvements beyond the classical theory of adt's.

In Sect. 3 we outline some relevant approaches; the presentation is related to the issues discussed in Sect. 2 and the specification examples presented in Sect. 1. However, being impossible to report on all methods, we have mainly used some approaches to illustrate, as examples, concrete ways of tackling the issues of the field..

1 Processes and Concurrent Systems

Informally a process is an entity with the capability of performing an activity within which it may interact with other entities and/or with the environment. The interaction may consist in communicating, synchronizing, cooperating, acting in parallel, competing for resources with other processes and/or with the environment.

By a concurrent system we informally mean a process consisting of component processes that are operating concurrently.

We are of course interested in those aspects of processes that support the design and implementation of software systems. Thus we are looking for a formal support to the specification, programming, implementation and verification phases. For this it is crucial to have good models for processes.

Now, the usual formal model of a sequential software system (program) as input/output, or state to state, function is no longer adequate for processes. Moreover, to date no single model seems to capture all the relevant formal aspects of a process. Hence in the following we will briefly introduce the most significant formal models which have been used in connection with the algebraic specification of processes. In the meantime we introduce some terminology typical of concurrency, which will be useful in the sequel.

Warning. We are presenting basic models and not formalisms; for example, labelled transition systems (and variations) are the common basic model for different formalisms like CCS, CSP, MEIJE,Π-calculus, etc. Moreover some formalisms, notably the many variations of Petri nets, allow to represent different aspects of systems and provide a variety of basic models. However our aim here is only to give the non-expert in concurrency some very basic information, in order to understand the following presentation of specification formalisms. It is not at all an overview of existing formalisms in concurrency.