Lecture Notes in Computer Science 2900
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Michel Bidoit   Peter D. Mosses

CASL User Manual

Introduction to Using the
Common Algebraic Specification Language

With chapters by

Till Mossakowski, Donald Sannella,
and Andrzej Tarlecki

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Authors

Michel Bidoit
Laboratoire Spécification et Vérification, CNRS UMR 8643
École Normale Supérieure de Cachan
61, Avenue du Président Wilson, 94235 Cachan Cedex, France
E-mail: bidoit@lsv.ens-cachan.fr

Peter D. Mosses
University of Aarhus, BRICS and Department of Computer Science
Aabogade 34, 8200 Aarhus N, Denmark
E-mail: pdmosses@brics.dk

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): D.2.1, D.3.1, D.2, D.3, F.3

ISSN 0302-9743
ISBN 3-540-20766-X Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 10931011 06/3142 5 4 3 2 1 0

Preface

Casl, the *Common Algebraic Specification Language*, has been designed by CoFI, the *Common Framework Initiative* for algebraic specification and development. Casl is an expressive language for specifying requirements and design for conventional software. It is algebraic in the sense that models of Casl specifications are algebras; the axioms can be arbitrary first-order formulas.

This User Manual illustrates and discusses how to write Casl specifications.

Casl is a major new algebraic specification language. It has been carefully designed by a large group of experts as a general-purpose language for practical use in software development – in particular, for specifying both requirements and design. Casl includes carefully-selected features from many previous specification languages, as well as some novel features that allow algebraic specifications to be written much more concisely and perspicuously than hitherto. It may ultimately replace most of the previous languages, and provide a common basis for future research and development.

Casl has already attracted widespread interest within the algebraic specification community, and is generally regarded as a de facto standard. Various sublanguages of Casl are available – primarily for use in connection with existing tools that were developed in connection with previous languages. Extensions of Casl provide languages oriented toward development of particular kinds of software (reactive, concurrent, etc.)

Major libraries of validated Casl specifications are freely available on the Internet, and the specifications can be reused simply by referring to their names. Tools are provided to support practical use of Casl: checking the correctness of specifications, proving facts about them, and managing the formal software development process.

The companion CASL *Reference Manual* [20] provides full details of the CASL design, including its formal semantics.

After briefly reviewing the background of CoFI and CASL, and the underlying concepts of algebraic specification languages, this book introduces the potential user to the features of CASL mainly by means of illustrative examples. It presents and discusses the typical ways in which the language concepts and constructs are expected to be used in the course of building system specifications. Thus, the presentation focuses on *what* the constructs and concepts of CASL are *for*, and *how* they should (and should not) be used. These points are made as clear as possible by referring to simple examples, and by discussing both the general ideas and some details of CASL specifications.

Further chapters introduce the reader to the CASL Reference Manual, to some of the currently available CASL support tools, and to a couple of the CASL libraries of basic datatypes. A substantial case study of the practical use of CASL in an industrially-relevant context completes the material. Appendices provide a quick reference of CASL constructs, a list of the main points to bear in mind when using CASL, and the original informal requirements for the case study.

Structure

Part I: Background

Chapter 1 describes the origins of CASL: how CoFI was formed in response to the proliferation of algebraic specification languages in the preceding two decades, and the aims and scope that were formulated for this international initiative.

For the benefit of readers not already familiar with other algebraic specification languages, Chap. 2 reviews the main concepts of algebraic specification, explaining standard terminology regarding specification language constructs and models (i.e., algebras).

Part II: Writing CASL Specifications

Chapter 3 shows how some familiar datatypes involving total functions are specified in CASL, essentially as in many other algebraic specification languages. Loose, generated, and free specifications are discussed in turn, with illustrative examples and advice on the use of different specification styles.

Partial functions arise naturally. Chapter 4 explains how CASL supports specification of partial functions, drawing attention to where special care is needed compared to specifications involving only total functions.

Subsorts and supersorts are often useful in CASL specifications. Chapter 5 illustrates how they can be declared and defined, and that they can sometimes be used to avoid the need for partial functions.

The examples given so far make use of named and structured specifications in a simple and natural way. Chapter 6 takes a much closer look at the constructs Casl provides for structuring specifications, explaining how large and complex specifications are easily built out of simpler ones by means of a small number of specification-building operations.

Chapter 7 shows how making a specification generic (when appropriate) improves its reusability, allowing it to be instantiated with different arguments; compound identifiers avoid the need for explicit renaming when combining the results of different instantiations. It also introduces the constructs for expressing so-called views between specifications.

While specification-building operations are useful to structure the text of large specifications, architectural specifications are meant for imposing structure on implementations. Chapter 8 discusses and illustrates the role of architectural specifications, and shows how to express them in Casl.

Chapter 9 explains and illustrates how libraries of named specifications can be formed, and made available over the Internet, to encourage widespread reuse and evolution of specifications. Version control is of crucial importance here.

Part III: Carrying On

Chapter 10 gives a detailed overview of the foundations of Casl, which are established in the accompanying Casl Reference Manual.

Tool support is vital for efficient use of formal specifications in connection with practical software design and development. Chapter 11 presents the main tools that have been implemented so far; several of them allow use of Casl specifications in connection with tools that were originally developed for other specification languages, showing how Casl provides tool interoperability.

Chapter 12 introduces a few of the many specifications that are available in the Casl libraries of basic datatypes.

Finally, Chap. 13 gives a realistic case-study of the use of Casl in practice, in connection with the design of software for a Steam-Boiler Control System. This particular example is one of the standard bench-marks for comparing specification frameworks [1].

Appendices and Indexes

This volume is completed by three appendices: App. A provides a compact overview of all Casl constructs, for quick reference; App. B lists all the main points to bear in mind when using Casl; and App. C reproduces the informal requirements specification for the case study.

The names of all the specifications given in this book are listed at the back, together with an index of concepts and a list of references to the literature. (A comprehensive annotated bibliography of publications involving Casl is provided in the Reference Manual.)

An accompanying CD-ROM contains the source files for all the illustrative specifications, and a copy of the libraries of specifications of basic datatypes.

Organization

All the main points are highlighted like this.

The material in this book is organized in a tutorial fashion. Each main point is usually accompanied by an illustrative example of a complete Casl specification; the names of these specifications are listed (both in order of presentation and alphabetically) at the end of the book. Moreover, the points themselves are repeated (in order of presentation) in App. B.

Readers who are familiar with previous algebraic specification languages, and especially those who have been following or participating in the design and development of Casl, may prefer to skip lightly through Chaps. 1 and 2. Chapter 3, however, is mandatory, since it is there that many Casl features needed to understand the subsequent chapters are introduced.

In contrast, Chaps. 4 and 5 can be skipped at first reading if the reader is not so much interested in partial functions, resp. subsorting (with the proviso though that there are some references to the examples given in these chapters from later chapters).

Chapters 6 and 7 present mainstream material, and until one feels comfortable with all the main points and examples, it is advisable to wait with proceeding to Chaps. 8 and 9.

Chapter 10 is primarily for those who will want to follow up on this book with a more detailed study of Casl, based on the Reference Manual. Part of Chap. 11 assumes familiarity with concepts introduced in Chap. 7. In Chap. 12, Sect. 12.1 assumes Chaps. 4 and 5, whereas Sect. 12.2 assumes also Chaps. 6 and 7. Finally, most of Chap. 13 can be studied after Chap. 7, but Sect. 13.10 requires Chap. 8.

Acknowledgement. Chapter 10 was written by Donald Sannella and Andrzej Tarlecki. Chapter 11 was written by Till Mossakowski, with contributions from Mark van den Brand and Markus Roggenbach. Till Mossakowski also provided Chap. 12, based on libraries of Casl specifications developed at the Bremen Institute for Secure Systems in joint work with Markus Roggenbach and Lutz Schröder; and he checked the well-formedness of all the specifications in this book, using Cats, the Casl Tool Set.

Public drafts of this book were released in July and October 2003. The many insightful comments from CoFI participants and other readers were very helpful to the authors during the preparation of the final version. Detailed comments on all or part of the public drafts were received from Jørgen Iversen, Christian Maeder, Guillem Marpons, Narciso Martí-Oliet, Till Mossakowski, Don Sannella, Giuseppe Scollo, Andrzej Tarlecki, Frédéric Voisin, and Alexandre Zamulin. Responsibility for any mistakes in the final version belongs, of course, to the authors.

Michel Bidoit gratefully acknowledges support from LSV,[1] CNRS, and École normale supérieure de Cachan. Peter Mosses gratefully acknowledges support from BRICS[2] and the Department of Computer Science, University of Aarhus. Much of the material on which this book is based was developed in connection with activities of CoFI-WG (ESPRIT Working Group 29432) and IFIP WG 1.3 (Foundations of System Specification).

Finally, special thanks to Springer, and in particular to Alfred Hofmann as Executive Editor, for their willingness to publish this book, and for helpful advice concerning its preparation.

Michel Bidoit and Peter D. Mosses

October, 2003

[1] Laboratoire Spécification et Vérification (`www.lsv.ens-cachan.fr`).
[2] Basic Research in Computer Science (`www.brics.dk`), funded by the Danish National Research Foundation.

Contents