1

Introduction

This chapter first explains the background and aims of CoFI, the Common Framework Initiative for algebraic specification and development of software. It then gives an overview of the main features of CASL, the Common Algebraic Specification Language.

1.1 CoFI

In 1995, an open collaborative effort was initiated: to design a common framework for algebraic specification and development of software. It is referred to as *The Common Framework Initiative*, CoFI.[1]

> *There was an urgent need for a common framework.*

The rationale behind this initiative was that the lack of such a common framework was a major hindrance for the dissemination and application of algebraic specification techniques. In particular, there was a proliferation of languages – some differing in only quite minor ways from each other. The major languages included ACT ONE/ACT TWO [19], ASF [6], ASL [36], CLEAR [15], EXTENDED ML [26], LARCH [25], OBJ3 [24], PLUSS [9], and SPECTRUM [14]. This abundance of languages was an obstacle for the adoption of algebraic methods for use in industrial contexts, making it difficult to exploit standard examples, case studies and training material. A common framework, with widespread support at least throughout the research community, was urgently needed.

[1] CoFI is pronounced like 'coffee'.

M. Bidoit and P.D. Mosses: CASL User Manual, LNCS 2900, pp. 3–9, 2004.

The aim of CoFI was to base the common framework as much as possible on a critical selection of features that had already been explored in previous research and applications (see the IFIP State-of-the-Art Report on *Algebraic Foundations of Systems Specification* [3] for the background). The collective experience and expertise of the CoFI participants provided a unique opportunity to achieve this aim within a reasonably short time-span.

The various groups working on algebraic specification frameworks had already had ample opportunity to develop their own particular variations on the theme of algebraic specification [16], yet no clear 'winner' had emerged (although there were several strong contenders).

CoFI aims at establishing a wide consensus.

The aim of CoFI was to design a framework incorporating just those features for which there would be a wide consensus regarding their appropriateness. This framework should be able to subsume many of the existing frameworks, and be seen as an attractive common basis for future research and development – with high potential for strong collaboration between the various groups.

The initial overall aims of CoFI were formulated as follows:

- A common framework for algebraic specification and software development is to be designed, developed, and disseminated.
- The production of the common framework is to be a collaborative effort, involving a large number of experts (30–50) from many different groups (20–30) working on algebraic specification.
- In the short term, the common framework is to become accepted as an appropriate basis for a significant proportion of the research and development in algebraic specification.
- Specifications in the common framework are to have a uniform, user-friendly syntax and straightforward semantics.
- The common framework is to be able to replace many existing algebraic specification frameworks.
- The common framework is to be supported by a concise reference manual, user's guide, libraries of specifications, tools, and educational materials.
- In the longer term, the common framework is to be made attractive for use in industrial contexts.
- The common framework is to be available free of charge, both to academic institutions and to industrial companies. It is to be protected against appropriation.

The focus of CoFI is on algebraic *techniques.*

The functionality of the common framework is to allow and be useful for:

- algebraic specification of the functional requirements of software systems, for some significant class of software systems;
- formal development of design specifications from requirements specifications, using some particular methods;
- documenting the relation between informal statements of requirements and formal specifications;
- verification of correctness of development steps from (formal) requirements to design specifications;
- documenting the relation between design specifications and implementations in software;
- exploration of the (logical) consequences of specifications: e.g., rewriting, theorem-proving, prototyping;
- reuse of parts of specifications;
- adjustment of specifications and developments to changes in requirements;
- providing a library of useful specification modules; and
- providing a workbench of tools supporting the above.

CoFI has already achieved its main aims.

The first major achievement of CoFI was the completion of the design of CASL, the Common Algebraic Specification Language. The CASL design effort started in September 1995. An initial design was proposed in May 1997 (with a language summary, abstract syntax, and formal semantics, but no concrete syntax) and tentatively approved by IFIP WG1.3. The report of the IFIP referees on the initial CASL design proposal suggested reconsideration of several points in the language design, and requested some improvements to the documents describing the design. Apart from a few details, the design was finalized in April 1998, and CASL version 1.0 was released in October 1998. IFIP WG 1.3 was asked to review the final design of CASL version 1.0.1 in May 2000, and subsequently approved that design in April 2001. The current version (1.0.2) was adopted in October 2003; it incorporates adjustments to some minor details of the concrete syntax and the semantics. No further revisions of the CASL design are anticipated.

The CASL *Reference Manual* [20], published as a companion volume to the present book, includes a detailed yet concise (60 pages) summary of the CASL design; the rest of it is concerned mainly with the formal syntax and semantics of CASL, and with the libraries of basic datatype specifications. An introduction to the Reference Manual is given in Chap. 10 of this book.

In parallel with the design of CASL, CoFI has developed tool support for the use of CASL (see Chap. 11), and substantial libraries of CASL specifications (see Chap. 12).

Despite the previous lack of a CASL User Manual, there is already much evidence that CASL is now accepted as an appropriate basis for research and development in algebraic specification. Reference [33] gives an overview of what was achieved in the period 1998–2001, and the annotated bibliography in the CASL Reference Manual lists a significant number of further publications that involve CASL. At the time of writing, it remains to be seen whether significant industrial take-up will follow.

CoFI is an open, voluntary initiative.

CoFI was started by COMPASS (ESPRIT Basic Research WG 3264/6112, 1989-96), in cooperation with IFIP WG 1.3 (Foundations of System Specification, founded 1992), on the basis of proposals made during their meetings in 1994 (Santa Margherita Ligure, Italy) and 1995 (Oslo, Norway); participation in CoFI was, however, never confined to members of those working groups. The active participants have included some 30 leading researchers in algebraic specification, with representatives from almost all the European groups working in this area. (Ideally, representatives from non-European groups would have been involved too, but logistic difficulties prevented this.)

Originally, CoFI had separate task groups concerning language design, semantics, tools, methodology, and reactive systems. There was a substantial amount of interaction between the task groups, which was facilitated by many of the CoFI participants being involved in more than one task group. The overall coordination of these task groups was managed by Peter Mosses from the start of CoFI in September 1995 until August 1998, and subsequently by Don Sannella. In 2003, the CoFI task groups were replaced by a looser coordination mechanism, with a steering committee chaired by Don Sannella.

CoFI has received funding as an ESPRIT Working Group, and is sponsored by IFIP WG 1.3.

The European Commission provided funding for the European component of CoFI as ESPRIT Working Group 29432 from 1998 to 2001 [33]. The partners were the coordinating sites of the various CoFI task groups (University of Aarhus, University of Bremen, École normale supérieure de Cachan, University of Genova, INRIA Lorraine, Warsaw University) with the University of Edinburgh as overall coordinator. The goals of the working group were to coordinate the completion of and disseminate the Common Framework, to

demonstrate its practical applicability in industrial contexts, and to establish the infrastructure needed for future European collaborative research in algebraic techniques. Apart from this period of funding, and support for meetings from the COMPASS Working Group until its termination in 1996, CoFI has relied entirely on unfunded efforts by its participants. Participation in the frequent working meetings was often supported by generous subsidies from the local organizers.

IFIP WG 1.3 sponsors CoFI by reviewing proposals for changes to the design of CASL, and proposals for extensions of CASL. Moreover, a considerable number of members of IFIP WG 1.3 have been (and, at the time of writing, still are) active participants of CoFI.

New participants are welcome!

CoFI is an open collaboration, and new participants are always welcome. Current information about CoFI activities is available at the main CoFI web site: `http://www.cofi.info`. The low-volume mailing list `cofi@cofi.info` is reserved for CoFI announcements, and discussions about CoFI activities generally take place on the mailing list `cofi-discuss@cofi.info`; see the CoFI web site for how to subscribe, and for access to the archives.

1.2 CASL

CASL has been designed as a general-purpose algebraic specification language, subsuming many existing languages.

The primary specification language developed by CoFI is called CASL: the Common Algebraic Specification Language. Its main features are:

- The design of CASL is based on a critical selection of the concepts and constructs found in existing algebraic specification frameworks.
- CASL is an expressive specification language with simple semantics and good pragmatics.
- CASL is appropriate for specifying requirements and design of conventional software packages.
- CASL is at the heart of a coherent family of languages that are obtained as sublanguages or extensions of CASL.

CASL subsumes many previous languages for the formal specification of functional requirements and modular software design. Tools for CASL are interoperable, i.e., capable of being used in combination rather than in isolation. CASL interfaces to existing tools extend this inter-operability (see Chap. 11).

The intention was to base the design of CASL on a critical selection of concepts and constructs from existing specification languages. However, it was not easy to reach a consensus on a coherent language design. A great deal of careful consideration was given to the effect that the constructs available in the language would have on such aspects as the methodology and tools. A complete formal semantics for CASL was produced in parallel with the later stages of the language design, and the desire for a relatively straightforward semantics was one factor in the choice between various alternatives in the design.

CASL represents a consolidation of past work on the design of algebraic specification languages. With a few minor exceptions, all its features are present in some form in other languages, but there is no language that comes close to subsuming it. Designing a language with this particular novel collection of features required solutions to a number of subtle problems in the interaction between features. An overview of the CASL design is presented in [2], and full details are provided in the CASL Reference Manual [20].

CASL is at the center of a family *of languages.*

It was clear from the start that no single language could suit all purposes. On the one hand, sophisticated features are required to deal with specific programming paradigms and special applications. On the other hand, important methods for prototyping and reasoning about specifications only work in the *absence* of certain features: for instance, term rewriting requires specifications with equational or conditional equational axioms.

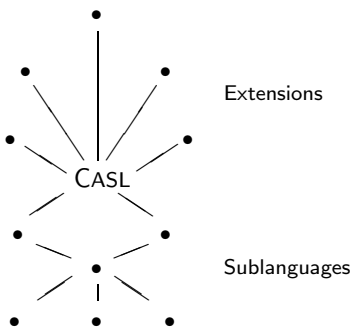

Fig. 1.1. The CASL Family of Languages

CASL is therefore at the center of a *family* of languages, see Fig. 1.1. Some tools will make use of well-delineated *sublanguages* of CASL, obtained by syntactic or semantic restrictions [29], while *extensions* of CASL are generally

designed to support various paradigms and applications. The design of CASL took into account the need to define sublanguages and extensions.

CASL itself has several major parts.

The major parts of CASL are concerned with *basic* specifications, *structured* specifications, *architectural* specifications, and *libraries* of specifications. They have been designed to be used together: basic specifications can be used in structured specifications, which in turn can be used in architectural specifications; structured and/or architectural specifications can be collected into libraries. However, these parts of CASL are quite independent, and may be understood separately, as we shall see in Part II.