An Algebraic Specification of the Steam-Boiler Control System

Michel Bidoit [1], Claude Chevenier [2], Christine Pellen [2] and Jérôme Ryckbosch [2]

[1] LIENS, C.N.R.S. U.R.A. 1327 & Ecole Normale Supérieure
45 Rue d'Ulm, F–75230 Paris Cedex 05, France

[2] Direction des Etudes et Recherches
Electricité de France

Abstract. We describe how to derive an algebraic specification of the Steam-Boiler Control System starting from the informal requirements provided to the participants of the Dagstuhl Meeting *Methods for Semantics and Specification*, organized jointly by Jean-Raymond Abrial, Egon Börger and Hans Langmaack in June 1995. The aim of this formalization process is to analyze the informal requirements, to detect inconsistencies and loose ends, and to translate the requirements into a formal, algebraic, specification. During this process we have to provide interpretations for the unclear or missing parts. We explain how we can keep track of these additional interpretations by localizing very precisely in the formal specification where they lead to specific axioms. Hence we take care of the traceability issues. We also explain how the formal specification is obtained in a stepwise way by successive refinements. Emphasis is put on how to specify the detection of the steam-boiler failures. Finally we discuss validation and verification issues. For this case study we use the PLUSS algebraic specification language and the LARCH PROVER.

Evaluation and Comparison

In this paper we have shown how to derive an algebraic specification of the Steam-Boiler Control System starting from the informal requirements. In particular, we have shown how this formalization process was useful to detect inconsistencies and loose ends in the informal requirements, and how to keep track in the formal specification of the interpretations we had to make in order to overcome the unclear or missing parts of the requirements. An interesting aspect of this case study is the demonstration of a method for deriving the formal specification in a stepwise way by successive refinements. We hope we have convinced the reader that the resulting specification is both clear and rather easy to understand. To achieve these aims we have used the PLUSS algebraic specification language which proves useful not only to express specifications in a structured and modular way, but also to make explicit the development process of the whole specification. An important part (although unfortunately too often neglected) of the specification task is the **validation** of the formal specification with respect to the informal requirements. One of the main advantages of the (classical) algebraic framework is that it allows us to benefit from the most advanced existing automated tools, and in particular we have shown how to use the LARCH PROVER to validate our algebraic specification. The importance of this validation task (and the suitability of the LARCH PROVER) was clearly demonstrated in our case study, and in particular we have shown that thanks to the proofs performed, we have detected a flaw in our specification (that was easily corrected).

Hence in this paper we have provided a formal requirements specification of the control program. Moreover, various safety properties are studied and proven formally. We have not specified the initialization phase of the Steam-Boiler Control System due to lack of time. However, it is obvious that the method used in the paper will apply to the specification of the initialization phase as well, and even if it is clear that the resulting specification would be a bit more complex (due to the intricacies of the initialization phase), there is absolutely no problem extending our specification by the description of the initialization phase. According to the work plan described in the introduction, we should in principle have derived from our requirements specification a design specification and finally a program. Again, this was not done due to lack of time, but we would like to stress that this second part of the work plan, although technical, is not really difficult. Our requirements specification is very detailed and contains a precise specification of the control program including specifications for the detection of equipment failures. Hence much of what would go into a functional design in other approaches is already contained in our requirements specification. Similarly, we have not provided a separate architectural design, but our algebraic requirements specification is structured in such a way that part of the architecture of the program will be derived from it quite easily.

The solution proposed by M.-C. Gaudel, P. Dauchy and C. Khoury (see Chap-

ter GDK, this book) is very similar to ours, the main difference being that they use a slightly different variant of algebraic specifications with implicit states.

We believe algebraic specifications provide just the right trade-off between expressive power and (relative) simplicity to specify systems like the Steam-Boiler Control System. Clearly, some expertise and training is required to *write* the algebraic specification (we believe height weeks training should be enough for an average programmer familiar with the basic concepts of logic), but we would like to emphasize the fact that almost no prior knowledge of algebraic specifications is required to *understand* it (only some basic background in logic is required). We estimate that the global time spent to write and validate the specification is one man-month (this does not include the time required to write the paper). Our claim that our algebraic specification of the Steam-Boiler Control System can be understood by people not trained in algebraic specifications is supported by concrete experience with some Electricité de France engineers.

This article was processed using the LaTeX macro package with LLNCS style