Abstract. As part of a comprehensive design concept for complex reactive systems we investigate the derivation of formal requirements and design specifications at systems level. We discuss the meaning of correctness with respect to the embedding of mathematical models into the physical world. A crucial aspect in our attempt to make the logic link between the application domain specific view and the formal view explicit is the concept of evolving algebra [13, 14]; it provides the formal basis of a specification methodology which has successfully been applied to a variety of specification and verification problems. We introduce an evolving algebra abstract machine as a conceptual framework for the development of tools for machine based analysis and execution of evolving algebra specifications.

1 Introduction

Reactive systems\(^1\) cover a wide range of challenging applications of computer based information, communication, and control systems; this class, in particular, does also include various kinds of concurrent systems such as parallel, distributed, or real-time systems, which by their nature are reactive rather than transformational. Reactivity thereby becomes the predominating behavioural characteristic of a system: the purpose of a system run is not necessarily to compute a final result, but to maintain some ongoing interaction with the external system environment.

Since the embedding of a reactive system into a given environment may impose considerable functional requirements having a strong impact on the entire design, a precise interface specification should be considered as the starting point for the design process. The interface specification explicitly describes how the system is influenced by its environment. Such an approach corresponds to an open system view, which is in contrast to a closed world assumption (closed system view) where everything relevant is included in the system [10]. According to widely accepted principles of systems engineering, for instance, the concept of model-based system design as stated by Wayne Wymore in his book on model-based systems engineering [24]:

\(^1\) We refer to the notion as defined by D. Harel and A. Pnueli in [16] and [22].
To design a system is to develop a model on the basis of which a real system can be built, developed, or deployed that will satisfy all its requirements.

the functional system requirements should be specified as complete, precise, and unambiguous as possible. Such a behavioural description requires to have a well-defined mathematical model as rigorous formalization of the functional requirements. However, a formal basis alone does not guarantee that the model fulfills the requirements; even if the model has been proven to be correct this does not necessarily mean that it really fits into the given environment, since the assumptions on which such proves are based may be incomplete or wrong. Practical experiences with applications of formal methods, like the “Production Cell” case study [20], have indeed shown that such mismatchings do frequently occur.

As part of a comprehensive design concept for complex reactive systems we investigate the derivation of formal requirements specifications from given informal system descriptions. More precisely, we are concerned with the formal definition of functional and timing requirements at the systems level. In our terminology which is based on the notion of evolving algebra, as described in the next section, the resulting mathematical models are called ground models.

The evolving algebra notation enables a specification style which makes the relationship between the informal description and the ground model explicit; this seems to be a crucial issue in any attempt at mathematical modeling of non-mathematical reality. A ground model should reflect our intuitive understanding of the intended system behaviour not only with the necessary completeness and precision, but also in a clear and understandable way. The ground model thus provides the logic link between the application domain specific view and the formal view.

After a brief introduction of the specification language in Section 2, we investigate the derivation of ground models in Section 3. To illustrate the concepts, Section 4 presents an example of specifying reactive system behaviour. In Section 5 we consider the development of tools for analysis and execution of specifications in a CAST context. Section 6 contains the conclusions.

2 Theoretical Foundations

The concept of evolving algebra\(^2\) as defined by Yuri Gurevich in [13, 14] provides the formal basis of our specification methodology. Evolving algebras are abstract

\(^2\) Correctness here can not have an absolute meaning (since there is no way to prove it), rather it means correctness with respect to certain assumptions about the physical world (as represented by the external environment).

\(^3\) The term algebra, as we use it here, is characterized by the notion of first-order structure of classical logic. A regular first-order structure consists of domains, functions, and relations, while structures without relations are called algebras [12]. (Note that relations can also be expressed through their characteristic functions.)