Advanced
Component Interface Specification\textsuperscript{1)}

Manfred Broy
Institut für Informatik
Technische Universität München
80290 München, Germany

Abstract

We introduce a method for the specification of reactive asynchronous components with a concurrent access interface and outline its mathematical foundation. The method supports the specification of components that show a complex reactive behavior including timing aspects. Examples are the nonstrict fair merge or the arbiter. The method supports the specification of reactive systems and their modular composition into data flow networks. The specification approach is compositional. It supports the integrated specification and verification of both safety and liveness conditions in modular system descriptions. We outline particular specification styles that may be useful for the better readability of such specifications.

1. Introduction

A descriptive functional semantic model of distributed systems of interacting components is of major interest in many research areas and applications of computing science and systems engineering. For the modular systematic development of systems we need precise and readable interface descriptions of system components. We require that such interface descriptions contain all informations about its syntactic and semantic properties needed in order to use it properly. In the interface specification we also describe the time dependency of a component.

Although ignored in theoretical computer science for a while, the incorporation of time and its formal representation is of essential interest for system models. In time dependent systems, the timing and the data values of the output depend upon the timing and the data values of the input. However, for certain components the timing of the input does not influence the data values of the output, but only their timing. Then we can describe the input/output behavior of a component without explicit reference to time.

\textsuperscript{1)} This work was partially sponsored by the Sonderforschungsbereich 342 "Werkzeuge und Methoden für die Nutzung paralleler Rechnerarchitekturen" and the industrial research project SysLab.
We are interested in the description of components that react by output to input. Both input and output takes place within a frame of time. When dealing with the time properties of systems we distinguish three basic classes of components characterized by their time dependencies:

- **time independence**: the data output values of a component do not dependent on the timing of the input data but only on their values,

- **weak time dependency**: some of the nondeterministic decisions in the behavior of a nondeterministic component is controlled by the timing of the input such as the relative order of messages on different input channels or the relative order of messages on input and output channels. But this is done in a superficial way that does not allow control of the nondeterministic decisions in the behavior (the output) by the quantitative timing of the input,

- **strong time dependency**: the behavior (the timing as well as the data output) depends on the quantitative timing of the input data.

The class of weak time dependent components brings specific problems for their functional specification. Examples are fair nonstrict merge or the arbiter (see section 4.2 and 6.3 for a detailed treatment). This has already been observed in [Park 83].

The modular specification of the observable behavior of interactive systems is an important technique in system and software development. We speak of black box specification or interface specification. An adequate concept of interface specification does not only depend on a simple notion of observability, but also on the operators that we apply to compose components into systems.

In the following we introduce a semantic model of interface behavior and study composition operators. On the basis of this semantic model we introduce more pragmatic specification techniques for the description of reactive components.

The paper is divided into a more theoretical and a more practical part. In the theoretical part we introduce and discuss the mathematical foundations. In the more practical part we give examples of component specifications and specification styles.

In a first chapter we introduce our mathematical basis. Then we show how to describe the syntactic interfaces and the dynamic behaviors of interactive systems. We treat composition operators. Finally, we briefly demonstrate different specification styles by some examples.

### 2. Streams

In this section we introduce the basic mathematical concepts for the description of systems by functional techniques. We define the set of streams over a given set of messages. We introduce a number of functions on this set. Finally we define the notation of stream processing function.