Refinement of Time

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

Abstract

We introduce a mathematical model of the timed behaviour of components with streams as input and output using a hierarchy of timing concepts. We distinguish non-timed streams, discrete streams with discrete or with continuous time, and dense streams with continuous time. We introduce a notion of a timed system component and formulate requirements for the time flow. We show how to compose timed systems in a modular way. We show that the introduction of time into a system model as well as the change of the timing model in the system development process is a refinement step.

1. Introduction

Although time is an important issue for many information processing systems all the early attempts to provide logical, algebraic, or mathematical foundations for programming and system development tried to abstract entirely from timing issues. This is of course fine as long as we are only interested in sequential algorithms. However, looking at interactive systems, especially at reactive embedded systems, timing issues become crucial. The same holds for many application systems of today that have to react to time events. This is why we need well worked out timing models that are well suited for the more abstract techniques for the specification, verification, and refinement of systems.

A descriptive functional semantic model of distributed systems of concurrently interacting components is of major interest in many research areas and applications of computing science and systems engineering. For the modular systematic development of information processing systems we need precise and readable interface descriptions of system components. We require that such interface descriptions document all information about the syntactic and semantic properties of a component that we need to use it properly. In an interface specification we also describe the time requirements and dependencies in the behaviour of a component.

We are interested in the description of components that react interactively to input by output. Both input and output take place within a global time frame. The modular specification of the observable behaviour of interactive systems is an important technique in system and software development. We speak of black box specifications or interface specifications. An adequate concept of interface specification does not only depend on a

---

1) This work was partially sponsored by the Sonderforschungsbereich 342 "Werkzeuge und Methoden für die Nutzung paralleler Rechnerarchitekturen", by the BMBF project KorSys, and the industrial research project SysLab.
simple notion of observability, but also on the operators that we apply to compose components into systems.

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. Often then the timing is rather unimportant. In these cases, we can describe the input/output behaviour of a component without explicit reference to time.

It is one of the goals of this paper to show what consequences the introduction of time in a semantic model actually has. The semantics gets more robust since the flow of time leads to a explicit modelling of causality and thus to more realistic models of computations. As a consequence, fixpoint theory gets more straightforward and does not need any sophisticated theoretical concepts. This simplicity is lost, however, if we abstract away timing information completely or partially as it is done, for instance, in models of computations based on the idea of full synchrony. Here the lack of explicit causality leads into semantic pathologies.

In the following, we introduce a semantic model of interface behaviour that includes discrete and continuous streams with discrete and continuous time and study composition operators. On the basis of this semantic model we introduce more pragmatic specification techniques for the description of reactive components.

In a first section, we introduce our mathematical basis. Then we show how to describe the syntactic interfaces and the dynamic behaviours of interactive systems. We treat composition operators and introduce concepts of refinement for timing issues.

2. Streams

In this section we introduce the basic mathematical concept of streams. A stream describes the communication history of a channel or the flow of values assumed by a variable of a system.

Roughly speaking a stream is a finite or an infinite sequence or flow of values from a set $M$ called the sort of the stream. If additional time information is contained, we speak of a timed stream. In the following, we are interested to separate aspects of data and message flow of a channel on variable from timing aspects. We define four concepts of streams: non-timed streams and discrete streams with discrete and continuous time, and finally dense streams with continuous time.

2.1 Non-timed Discrete Streams

Given a set $M$ of messages a discrete stream over the set $M$ is a finite or infinite sequence of elements from $M$. By $M^*$ we denote the finite sequences over the set $M$. The set $M^*$ includes the empty sequence that we denote by $\epsilon$. By $s^r$ we denote the concatenation of two sequences $s$ and $r$.

By $M^{\infty}$ (we write $M^{\infty}$ for the function space $\mathbb{N}^+ \rightarrow M$ where $\mathbb{N}^+$ stands for $\mathbb{N} \setminus \{0\}$) we denote the set of infinite sequences over the set $M$. By $M^{\infty}$ we denote the set of non-timed streams. It is defined by