Duration Calculi: An Overview
(Extended Abstract)

ZHOU Chaochen*

Abstract. The Duration Calculi are calculi for designing real-time software embedded systems. The Calculi overviewed in the paper include the Duration Calculus, the Extended Duration Calculus, the Mean Value Calculus and the Probabilistic Duration Calculus. They are extensions of an Interval Temporal Logic, and can accommodate some concepts of mathematical analysis such as integrals, mean values, piecewise continuity and differentiability of functions over continuous time. The Calculi can be used to capture and refine real-time requirements for hybrid systems, to define real-time behaviour/semantics of computing systems, and to calculate system dependability in terms of requirement satisfaction probability.

1 Summary

The research of the Duration Calculi was started by the ProCoS project (Provably Correct Systems - Esprit BRA 3104) in 1989, when the project was researching formal techniques for designing time/safety critical software embedded systems. Four calculi have been developed since then. They are the Duration Calculus, the Extended Duration Calculus, the Mean Value Calculus and the Probabilistic Duration Calculus.

The Duration Calculus is a real-time interval logic [14]. It formalises integrals of Boolean functions over intervals, and is used to specify and reason about timing and logical constraints on discrete states of a system. All the other calculi are extensions of the Duration Calculus. The Extended Duration Calculus [16] extends the Duration Calculus with piecewise continuity/differentiability of functions. It can capture properties of continuous states. This calculus is useful for reasoning about hybrid systems with a mixture of continuous and discrete states. The Mean Value Calculus [15] extends the Duration Calculus by replacing integrals of Boolean functions with their mean values, so that it can use δ-functions to represent instant actions such as communications and events. The Mean Value Calculus can be used to refine from state based requirements via mixed state and event specifications to event based specifications or programs. The Probabilistic Duration Calculus [5] [6] provides designers with a set of rules to reason about and calculate whether a given requirement written in the Duration Calculus will hold with a sufficient high probability for given failure

* UNU/IIST, P. O. Box 3058, Macau. E-mail: zcc@iist.unu.edu. On leave of absence from the Software Institute, the Chinese Academy of Sciences, Beijing, China.
probabilities of the components used in a design, where the design with imperfect components is taken to be a finite automaton with history-independent transition probabilities.

The Duration Calculi have been used to capture and refine requirements and designs for a number of examples, including a gas burner [9], a railway crossing [10], a water level controller [1] and an auto pilot [8]. The Calculi have also been used to define real-time semantics for Occam-like languages [12] [4], and to specify real-time behaviour of schedulers [12] and circuits [2].

In the paper we give an overview of the four duration calculi.

2 The Duration Calculus

A case study of the ProCoS project was asked to formulate a requirement for a gas burner:

"The proportion of time when gas is leak is not more than one twentieth of the elapsed time, if the system is observed for more than one minute."

The direct formulation of the requirement can be obtained by applying mathematical analysis. That is

\[(e - b) \geq 60 \sec \Rightarrow 20\int_b^e \text{Leak}(t) dt \leq (e - b)\]

where \(\text{Leak}\) is a Boolean function, which represents the leak state of the gas burner. The function is from reals \(\mathbb{R}\) (representing time) to \{0, 1\}, where 1 denotes that the system is in the state, and 0 denotes that the system is not in the state. The observation interval is taken to be bounded, and \(b\) stands for its start and \(e\) stands for its end.

Unfortunately, at that time, no software design calculi were available to express and reason about properties of integrals or differentials of functions, although integrals and differentials have been widely used for ages in control theory to model dynamic systems.

Confining ourselves to the formalisation of integrals of Boolean functions, we developed the Duration Calculus [14]. Integrals can be considered curried functionals from state functions and intervals to reals:

\[\int : S \rightarrow (I \rightarrow \mathbb{R})\]

where \(S\) stands for states (i.e. Boolean functions), and \(I\) for bounded intervals. Therefore the Interval Temporal Logic [7] (extended with continuous time) is adopted as its base logic, and interval functions \(\int S, \int P, \ldots\) become interval variables of the calculus, where \(S\) and \(P\) are states. It follows that \(\int 1 = e - b\), and we use \(l\) as an abbreviation of \(\int 1\) to be a mnemonic notation of interval length. Thus the requirement can be encoded more succinctly as \(\text{Req}\):

\[l \geq 60 \Rightarrow 20\int \text{Leak} \leq l\]

With integrals we can also express a lasting presence of a state over an interval by defining a ceiling operator \(\lceil . \rceil\):