Proved Development of the Real-Time Properties of the IEEE 1394 Root Contention Protocol with the Event B Method

Joris Rehm[*,***], Dominique Cansell[**,***] [1]

[*]Université Henri Poincaré Nancy 1
joris.rehm@loria.fr
[**]Université de Metz
cansell@loria.fr
[***]LORIA - BP 239 - 54506 Vandœvre-lès-Nancy - France

Abstract. We present a model of the IEEE 1394 Root Contention Protocol with a proof of Safety. This model has real-time properties which are expressed in the language of the event B method: first-order classical logic and set theory. Verification is done by proof using the event B method and its prover, we also have a way to model-check models. Refinement is used to describe the studied system at different levels of abstraction: first without time to fix the scheduling of events abstracly, and then with more and more time constraints.

1 Introduction

In this paper, we present a model of the IEEE 1394 Root Contention Protocol with a proof of safety and of real-time properties. We already described the pattern of our model of time, applied in a simple case study, in Cansell et al. (2007) as a pattern of refinement for the event B method. We show here how this pattern works over a proven development of the IEEE case study. Many different models for real-time already exist. Our goal is to find a model of time adapted to make proof by invariant with refinement over systems of events. We also argue that is better to start a proven development by an abstract model without time and to use refinement to add real-time properties. Therefore our model of time must allow us to use refinement.

The IEEE 1394, also known as FireWire, is used to connect devices like external hard-disks or movie cameras. Devices are able to configure themselves by the IEEE 1394 leader election protocol. This protocol takes the network as an acyclic graph and orients edges to obtain a spanning tree rooted by a leader. This general case is already done with the event B method in Abrial et al. (2003). This work extends this result to the following case: at the end of the algorithm, or when only two devices are connected, the general algorithm can fail. In this case the signals can cross in the bi-directional channel between the two devices if they send signals at almost the same send time. Consequently the IEEE 1394 Root Contention Protocol takes place in order to choose a leader between the two devices. The algorithm is probabilistic and uses a random choice between a short and a long waiting time. This sleeping time and signal sending between devices leads to a (probable) election. We can see this illustrated in

[1]This work was supported by grant No. ANR-06-SETI-015-03 awarded by the Agence Nationale de la Recherche.