Abstract. The B-Method is a state-based formal method that describes behaviour in terms of MACHINES whose states change under OPERATIONS. The process algebra CSP is an event-based formalism that enables descriptions of patterns of system behaviour. We present a combination of the two views where a CSP process acts as a control executive and its events simply drive corresponding OPERATIONS. We define consistency between the two views in terms of existing semantic models. We identify proof conditions which are strong enough to ensure consistency and thus guarantee safety and liveness properties.

Keywords: B-Method, CSP, Embedded Systems, Programming Calculi, Combining Formalisms.

1 Introduction

State based methods such as B specify functional aspects of a system and the effect of individual operations. On the other hand event-based process algebras are concerned with patterns of operations. System designers are interested in both these aspects of a system and thus a combination of state and event-based descriptions of a system is desirable. The systems that originally motivated our need to consider both viewpoints were safety-critical systems, for example embedded interlock systems. This paper provides a safe way of describing a combined view of a system.

Systems have successfully been modelled as collections of interdependent machines within the B Method. An abstract MACHINE is described using the Abstract Machine Notation (AMN). In this paper we adopt the convention that AMN keywords are indicated in italic capitals. Large MACHINEs can be constructed from other MACHINEs using INCLUDES, SEES and other constructs. A MACHINE encapsulates some local state and a collection of modules called OPERATIONS. OPERATIONS in a MACHINE can be pre-conditioned or guarded. We are interested in specifying embedded systems and refer to a B Abstract System in terms of the MACHINE at the top of the hierarchy of MACHINES which specify the following two kinds of OPERATIONS. Firstly, pre-conditioned OPERATIONS describe the modules which will be refined to code. They have the form PRE R THEN T END. If an OPERATION is invoked when the precondition $R$ is true it will behave as specified by $T$. However, if the OPERATION
is invoked outside its pre-condition the resulting execution may be an incorrect behaviour of the system. Secondly, the OPERATIONS which provide a model of the system context have the form SELECT P THEN V END, where P is a guard and V describes the effect of invoking the OPERATION. Guards are predicates on the state of a MACHINE which constrain the cases when an OPERATION is entitled to be invoked. If an OPERATION is invoked when the guard is true then the system will behave as expected with respect to the specification as was the case above. However, if the guard is false then execution is blocked.

Process algebras such as Communicating Sequential Processes (CSP) [8] are concerned with the evolution of systems as they execute sequences of events. They are appropriate for describing execution patterns. In this paper, we will show how events in a CSP recursive loop determine which corresponding OPERATION should execute. Thus we view the AMN specifications as providing abstract models of reactions to events. The recursive loop can be viewed as an execution checker and we will refer to it as a control executive. Thus in a combined view of a system a control executive for a system is described using a process algebra which in turn drives the individual state transitions of an Abstract System.

In general a CSP control executive could invoke an OPERATION outside its pre-condition, resulting in divergent behaviour. In [15] we gave conditions which ensured this did not occur. With guarded OPERATIONS we also need to ensure deadlock freedom so that a control executive never gets stuck trying to invoke OPERATIONS which are blocked. Ensuring deadlock freedom is the contribution of this paper.

The main result of this paper is that we introduce a new proof condition which guarantees deadlock freedom in the context of divergence freedom. Furthermore, we verify that this new condition is strong enough to ensure the consistency of a combined system consisting of guarded OPERATIONS. In this verification we think of an Abstract System as a process and its combination with the control executive is essentially their parallel composition in CSP.

In formally justifying the link between these state and event-based methods we were influenced by the existing correspondence between Action Systems and CSP. This correspondence is described by Morgan [9] in terms of weakest precondition semantics and the failures-divergences model.

We assume the reader is familiar with AMN. Further details can be found in [1]. However, we will introduce the CSP notation we require. This paper is organised as follows. Section 2 gives a brief overview of CSP. Sections 3, 4 and 5 contain the main contribution of the paper. They present the theoretical foundations of the specific relationship between B and CSP. Section 6 illustrates this new relationship in relation to our previous work on divergence freedom. The final section contains a discussion and conclusions. Proofs of the results have been omitted for reasons of space and can be found in the technical report [16].