Statecharts via Process Algebra*

Gerald Lüttgen¹, Michael von der Beeck², and Rance Cleaveland³

¹ Institute for Computer Applications in Science and Engineering
NASA Langley Research Center, Hampton, VA 23681-2199, USA
luettgen@icase.edu

² Department of Computer Science, Munich University of Technology
Arcisstr. 21, D-80290 München, Germany
beeck@in.tum.de

³ Department of Computer Science, State University of New York at Stony Brook
Stony Brook, NY 11794-4400, USA
rance@cs.sunysb.edu

Abstract. Statecharts is a visual language for specifying the behavior of reactive systems. The language extends finite-state machines with concepts of hierarchy, concurrency, and priority. Despite its popularity as a design notation for embedded systems, precisely defining its semantics has proved extremely challenging. In this paper, we present a simple process algebra, called Statecharts Process Language (SPL), which is expressive enough for encoding Statecharts in a structure-preserving and semantics-preserving manner. We also establish that the behavioral equivalence bisimulation, when applied to SPL, preserves Statecharts semantics.

1 Introduction

Statecharts is a visual language for specifying the behavior of reactive systems. The language extends the notation of finite-state machines with concepts of (i) hierarchy, so that one may speak of a state as having sub-states, (ii) concurrency, thereby allowing the definition of systems having simultaneously active subsystems, and (iii) priority, so that one may express that certain system activities have precedence over others. Statecharts has become popular among engineers as a design notation for embedded systems, and commercially available tools provide support for it. Nevertheless, precisely defining its semantics has proved extremely challenging, with a variety of proposals being offered for several dialects of the language. The semantic subtlety of Statecharts arises from the language’s capability for defining transitions whose enabledness disables other transitions. A Statechart may react to an event by engaging in an enabled transition, thereby

---

* This work was supported by the National Aeronautics and Space Administration under NASA Contract No. NASA1-97046 while the first author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-2199, USA. The third author was supported by NSF grants CCR-9257963, CCR-9505662, CCR-9804091, and INT-9603441, AFOSR grant F49620-95-1-0508, and ARO grant P-38682-MA.

performing a so-called \textit{micro step}, which may generate new events that may in turn trigger new transitions while disabling others. When this chain reaction comes to a halt, one execution step — also referred to as a \textit{macro step} — is complete. At a technical level, the difficulty for defining an operational semantics capturing the “macro-step” behavior of Statecharts arises from the fact that such a semantics should exhibit the following desirable properties: (i) the \textit{synchrony hypothesis} \cite{2}, which guarantees that a reaction to an external event terminates before the next event enters the system, (ii) \textit{compositionality}, which ensures that the semantics of a Statechart is defined in terms of the semantics of its components, and (iii) \textit{causality}, which demands that the participation of each transition in a macro step must be causally justified. Huizing and Gerth showed that an operational semantics in which transitions are labeled purely by sets of events — i.e., the “observations” a user would make — cannot be given, if one wishes all three properties to hold \cite{15}. In fact, the traditional semantics of Statecharts — as defined by Pnueli and Shalev \cite{22} — satisfies the synchrony hypothesis and causality, but is not compositional. Other approaches, e.g. \cite{17}, have achieved all three goals, but at the expense of including complex information regarding causality in transition labels.

While not as well-established in practice, \textit{process algebras} \cite{1,12,21} offer many of the semantic advantages that have proved elusive in Statecharts. In general, these theories are operational, and place heavy emphasis on issues of compositionality through the study of \textit{congruence relations}. Many of the behavioral aspects of Statecharts have also been studied for process algebras. For example, the synchrony hypothesis is related to the \textit{maximal progress assumption} developed in \textit{timed} process algebras \cite{11,28}. In these algebras, event transitions and “clock” transitions are distinguished, with only the latter representing the advance of time. Maximal progress then ensures that time may proceed only if the system under consideration cannot engage in internal computation. Clocks may therefore be viewed as “bundling” sequences of event transitions, which may be thought of as analogous to “micro steps,” into a single “time step,” which may be seen as a “macro step.” The concept of priority has also been studied in process-algebraic settings \cite{11}, and the Statecharts hierarchy operator is related to the \textit{disabling} operator of LOTOS \cite{3}.

In this paper, we present a new, \textit{process-algebraic} semantics of Statecharts. Our approach synthesizes the observations above; specifically, we present a new process algebra, called \textit{Statecharts Process Language (SPL)}, and we show that it is expressive enough for embedding several Statecharts variants. SPL is inspired by Hennessy and Regan’s \textit{Timed Process Language (TPL)} \cite{11}, which extends Milner’s CCS \cite{21} by the concept of an abstract, global clock. Our algebra replaces the handshake communication of TPL by a \textit{multi-event communication}, and introduces a mechanism to specify \textit{priority} among transitions as well as a \textit{hierarchy operator} \cite{25}. The operational semantics of SPL uses SOS rules to define a transition relation whose elements are labeled with simple sets of events; then, using traditional process-algebraic results we show that SPL has a compositional semantic theory based on \textit{bisimulation} \cite{21}. We connect SPL with Statecharts by