Verification of Quantitative Temporal Properties of SDL Specifications*

Iulian Ober[1,2] and Alain Kerbrat[2]

[1] Institut National Polytechnique/IRIT
2 Rue Camichel, 31000 Toulouse, France.
iulian.ober@enseeiht.fr
[2] Telelogic, 150 Rue N. Vauquelin, 31106 Toulouse, France.
{iulian.ober,alain.kerbrat}@telelogic.com

Abstract. We describe an approach for the verification of quantitative temporal properties of SDL specifications, which adapts techniques developed for timed automata [2]. With respect to other verification approaches applied to SDL, our approach broadens the class of analyzable specifications and improves the handling of non-deterministic systems, such as open systems communicating with an unspecified environment. Compared to the initial framework of timed automata, the application of these verification techniques to SDL raises two interesting issues, discussed in the paper. They are: expressing the semantics of time in SDL in terms of timed automata concepts, and employing a user friendly automata-based property specification language (GOAL [1]) to express and verify temporal properties. The paper also presents a verification tool prototype for SDL which implements these ideas.

1 Introduction

Automatic verification of behavioral properties is an important feature which can be offered by SDL tools, and which distinguishes SDL from other modeling languages used for specification and design of complex systems. Automatic verification is made possible by the existence of a formal semantics and syntax for the language [13], which can be used as a basis for mathematical reasoning about SDL specifications. Verification is an important task in the development of safety critical systems, which constitute an important share of the systems built with SDL.

Automatic verification capabilities implemented in industrial SDL tools (e.g. [19,17]), employ techniques derived from model checking. Properties which can be verified range from simple safety properties (e.g. absence of deadlocks, invariance of a logical condition) to more complex safety or liveness properties

* Work supported by the European project INTERVAL IST-1999-11557, *Formal Design, Validation and Testing of Real-Time Telecommunications Systems* (http://www.cordis.lu/ist/projects/99-11557.htm) and by the French RNRT Project PROUST (http://www-verimag.imag.fr/PROUST/).

R. Reed and J. Reed (Eds.): SDL 2001, LNCS 2078, pp. 182–202, 2001.

(e.g. linear properties specified by (Büchi) automata). The value added by SDL compared to other automatic verification approaches is twofold:

1. model checking is done on the analysis or design model (a functional model), without the need for building more abstract models in another formalism;
2. model checking is brought to the level of non specialists, by using simple and intuitive property specification languages (such as GOAL [1]) and tools.

As SDL is increasingly used for designing and implementing real-time and embedded systems, the verification methods and tools for SDL must be extended to account for quantitative temporal properties. This paper presents the results of a work aiming to adapt analysis techniques initially developed for timed automata [2] to SDL. This work also points out a series of deficiencies of SDL, which diminish its expressivity and the power of the analysis tools based on it. For the identified problems, we suggest some ways to improve the language.

The rest of the paper is structured as follows: Sect. 2 presents the state of the art in the specification and verification of temporal behavior using timed automata. We complete this section with a discussion of the limitations of timed automata, and their impact on our SDL approach. In §3 we show how the SDL execution model relates to the timed automata model. We also discuss possible improvements of SDL suggested by this relation. In Sect. 4 we examine how GOAL [1] can be used to describe temporal properties of SDL systems. In Sect. 5 we present the temporal verification facilities implemented in an extended version of the *Object*GEODE tool [19]. Finally, we draw the conclusions of this work and examine further advancement possibilities.

1.1 Related Work

The idea of applying timed automata techniques in higher level formalisms was intensively studied during the past few years. SDL is not the only language to benefit from this trend: proposals for the improvement of both formal description techniques (LOTOS, ESTELLE) and more informal modeling languages (UML) using timed automata are being studied.

For relating SDL and timed automata, [5] proposes an intermediate formalism, called IF, in which one can describe a system of asynchronously communicating processes much in the same way as in SDL. The authors propose a methodology in which SDL specifications are translated to IF, and IF models are analyzed using timed automata methods and tools.

[8] presents an approach in which a variant of timed automata, called Timed Finite State Machines (TFMS), are used as an intermediary form for generating SDL specifications and timed test cases from timed scenario descriptions. Timed automata techniques are not used for verification of properties, but the authors point out some of the same deficiencies of SDL that we reveal in this paper.

The general problems with expressing time-related behavior, discovered during this work and the work of our project partners are systematized in [6].