1 Introduction

IF\cite{5, 6, 10, 15} is an open validation platform for asynchronous timed systems such as telecommunication protocols or distributed applications developed at Verimag during the last 5 years.

The toolbox is built upon an specification language based on timed automata extended with discrete data variables, various communication primitives, dynamic process creation and destruction. This language is expressive enough to represent most useful concepts of modeling and programming languages for distributed systems (like SDL, UML, Java,...)

The core of the toolbox consists of a set of model-based validation components including exhaustive/interactive simulation, on-the-fly temporal logic model-checking, test case generation and optimal path extraction. In order to control state explosion, the toolbox provides several static analysis tools operating at the source level such as live variable analysis, dead-code elimination and slicing \cite{3, 4}. Finally, the toolbox is connected to commercial environments (such as Rational Rose, Rhapsody, Objecteering, Object Geode) and may be used for validating SDL and UML specifications \cite{7, 8, 11, 14}.

The toolbox has been successfully applied on several case studies including telecommunication protocols, distributed algorithms, real-time controllers, manufacturing, asynchronous circuits \cite{9, 13, 12, 2, 1}.

2 Objectives

The objectives of this tutorial are the following:

- first, it will give a complete presentation of the main functionalities of the IF validation environment, and
second, it will show how this environment can be used to experiment on new model-checking techniques.

Expected attendees are people interested in model-checking techniques, either from an (experienced) user or from tool designer or researcher point of view.

3 Summary of Material

In this tutorial, we will guide participants through the concepts and the use of the IF language and the associated tools. More precisely, we will focus on the following items:

language

In the first part we will provide a survey of the main concepts of the IF language. We will focus on both functional features (structure, communication, dynamic creation, external code integration) and non-functional ones (real-time primitives, resource management, priorities). Moreover, we will show how to express properties on IF specifications by means of dedicated observers.

core tools

In this second part we will introduce the toolbox architecture and its main components. We will describe the two main APIs: the syntax level API (abstract syntax tree) and the semantic level API (state graph). Among the tools, we will focus on the static analyser and some of the model based tools (e.g., model checker, test generator, optimal path extractor).

front-ends and applications

Finally, the third part will be dedicated to existing front-ends to SDL and UML. It will also give an overview of the most relevant case studies handled with the IF toolbox.

The tutorial will be illustrated with examples, on-line demos and comparisons with other related tool environments (Spin, CADP, Kronos, Uppaal, etc). Participants will receive CDs with the latest version of the IF toolbox and an example repository including the examples used in the tutorial.

References


