Chapter 19
Industrial Experiments in IMS, ATC, and SDR Projects of Property Verification Techniques

Emmanuel Gaudin
PragmaDev, France

ABSTRACT

The increasing complexity of embedded systems calls for verification techniques to make sure the systems behave properly. When it comes to safety-critical systems, this aspect is even more relevant and is now taken into consideration by certification authorities. For that matter, property verification is accepted to be done not only on the system itself but also on a representative model of the system. This chapter first introduces the different properties and how they could be expressed. Then associated modeling languages characteristics are discussed to describe the systems on which the properties can be verified. Finally, different technologies to verify the properties are presented, including some practical examples and existing tools. This last part is illustrated by several research projects such as the PRESTO ARTEMIS European project and the exoTICus System@tic Paris Region competitiveness cluster project.

PROPERTY DEFINITION

Properties of a system are difficult to define because they either seem obvious, like a plane should fly, or very tricky to explain, like the minimum speed of the plane is relative to its height and weather condition. Generally speaking the properties can be expressed as a set of relations on the inputs and the outputs of the system, or as an evaluation of some internal data in the system. For example a property such as "the speed of the plane should not exceed 900km/h" could be expressed by a mathematical expression such as "system.speed<900" which can be verified automatically by a computer. Usually speaking properties are categorized to be functional or non-functional, and black box or white box.

Functional Property

A functionality is a given scenario of what a system should do; one of the expected behavior. If the system behaves differently it would not mean the functionality is not fulfilled, because all alterna-

DOI: 10.4018/978-1-4666-6194-3.ch019

tive scenarios are not meant to be described for a given functionality. But a functional property describes a mandatory relation between several events. It can be seen as an extension of a function of the system, a piece of scenario that should be verified in any case.

Non Functional Property

A non functional property is generally speaking about the quality level of a property. For example a basic functional property might be that when the button is pressed the light shall lit. A non functional requirement could be that the light should lit within 10mS after pressing the button. Or it could be that the light bulb should have a 10,000 hours lifetime. In a sense a functional property is very straightforward and can be mathematically verified where a non-functional property is more about the quality of a property and is difficult to evaluate. Still one of the verifiable non-functional properties is performance because it is easy to measure, and it can even be estimated with detailed models of the system.

Black Box Property

A black box property is a property that looks at the system as a black box and that only considers what happens on its interfaces. The interest of this type of property is that it can be verified on a real system or a simulated model of the system. The problem is that since the internal information is not visible, to make sure the property is always verified requires all possible surrounding situations to be generated. Without any knowledge of the internal design of the system, the number of possible scenarios might be so large that it might not be possible to explore them all.

White Box Property

A white box property will use the internal information of the system. It is therefore dependent on the system implementation. Verification of a white box property is by construction easier and more efficient but is not always applicable on a real system because the internal information might not be accessible.

TECHNOLOGIES

We will focus on event driven technologies regarding property verification. These technologies apply to communicating system where the possible number of values for the inputs in the system, and the possible sequences of events can create a huge number of different cases for the system. It is therefore interesting to use a model of the system so that the property can be verified on a simulated model.

Executable Model

A model is an abstract representation of the real system. A model is very useful for the different stakeholders to communicate, to document what is the system about, to ease maintenance and evolution of the system. A model can be very abstract or quite detailed depending on the goals to achieve.

In order to be able to verify properties which are essentially dynamic, the model must be somehow executable. That requires a semantic of execution and an action language in the modeling language. In the domain of communicating systems, a semantic of execution is about how the information is sent and received, how time is handled, and how concurrent processing is organized. An action language is a way to formally write precise instructions. To do so the modeling language requires data types and some basic operators. This is necessary even to express a simple behavior such as: the system, in case of no reply within 1s, sends again the same message three times. In the absence of a proper response the system goes to a special error state. These types of languages are called formal because they are complete and unambiguous. For the description of event driven systems the most mature language is certainly SDL