# Formal Testing of Timed and Probabilistic Systems

Manuel Núñez



# Formal Testing of Timed and Probabilistic Systems *

Manuel Núñez

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid, Madrid, Spain,
mn@sip.ucm.es

**Abstract.** This talk reviews some of my contributions on formal testing of timed and probabilistic systems, focusing on methodologies that allow their users to decide whether these systems are *correct* with respect to a formal specification. The consideration of time and probability complicates the definition of these frameworks since there is not an obvious way to define correctness. For example, in a specific situation it might be desirable that a system is as fast as possible while in a different application it might be required that the performance of the system is exactly equal to the one given by the specification. All the methodologies have as common assumption that the system under test is a black-box and that the specification is described as a timed and/or probabilistic extension of the finite state machines formalism.

## 1 Introduction

Testing was classically considered an informal discipline. Actually, it was assumed that formal methods and testing were orthogonal lines of research. Quoting from Edsger W. Dijkstra's ACM Turing Lecture:

> *Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence. The only effective way to raise the confidence level of a program significantly is to give a convincing proof of its correctness.*

However, early work already showed that it is possible to successfully combine formal methods and testing [21, 6] and during the last 20 years there has been a vast amount of work on formal testing [9]. In particular, several workshops are exclusively devoted to the topic and formal testing has a strong presence in testing conferences, including this one, and in scientific journals.

Formal testing techniques provide systematic procedures to check systems in such a way that the coverage of their critical parts/aspects depends less on the intuition of the tester. While the relevant aspects of some systems only concern

* This research has been partially supported by the Spanish MICINN project TESIS (TIN2009-14312-C02-01) and by the Santander-UCM Programme to fund research groups (GR35/10-A-910606).

*what* they do, in some other systems it is equally relevant *how* they do what they do. Thus, after the initial consolidation stage, formal testing techniques started to deal with properties such as the probability of an event to happen, the time that it takes to perform a certain action, or the time when a certain action happens.

The work on formal testing of timed systems has attracted a lot of attention during the last years. Most work considers that time is *deterministic*, that is, time requirements follow the form "after/before $t$ time units..." Even though the inclusion of time allows to give a more precise description of the system to be implemented, there are frequent situations that cannot be accurately described by using this notion of deterministic time. For example, in order to express that a message will arrive at any point of time belonging to the interval $[0, 1]$ we will need, in general, infinite transitions, one for each possible value belonging to the interval. In this case, it would be more appropriate to use time intervals to describe the system. Let us consider now that we have to simulate the performance of a petrol station. Since the arrival of cars follows a Poisson distribution, we would need again to use an infinite number of transitions. Moreover, if we have to use a time interval we would be very imprecise since all that we could say is that the next car will arrive in the interval $[0, \infty)$. Thus, it would be very useful to have a mechanism allowing to express that a time constraint is given by using a random variable that follows a precise probability distribution function.

In addition to consider the temporal behavior of systems, it is also interesting to study their probabilistic behavior. The use of probabilities allows to quantify the non-deterministic choices that a system may undertake. For example, instead of just specifying that a dice can non-deterministically return a value between 1 and 6, we can give more information by pointing out that the probability associated with each of these values is equal to $\frac{1}{6}$. In order to introduce probabilities, we consider a variant of the *reactive* model [7]. A reactive model imposes a probabilistic relation among transitions departing from a given state and labeled by the same action but choices between different actions are not quantified. In our setting, we express probabilistic relations between transitions outgoing from a state and having the same input action (the output may vary). Technically, for each input and state, the addition of the probabilities associated with transitions departing from a given state and labeled with that input is equal to either 0 (if there are no transitions) or to 1.

## 2 Outline of the talk

The bulk of the talk is devoted to present formal testing methodologies where the temporal behavior of systems is taken into account and it is based on a joint work with Mercedes G. Merayo and Ismael Rodríguez [16]. The last part of the talk shows how the timed framework can be extended with probabilistic information. This part of the talk is based on a joint work with Ana Cavalli, Iksoon Hwang and Mercedes G. Merayo [13].

During this talk, the considered formalisms are always simple extensions of the classical concept of *Finite State Machine*. Intuitively, transitions in finite state machines indicate that if the machine is in a state $s$ and receives an input $i$ then it will produce an output $o$ and it will change its state to $s'$. An appropriate notation for such a transition could be $s \xrightarrow{i/o} s'$. If we consider a timed extension of finite state machines, a transition such as $s \xrightarrow{i/o}_t s'$ indicates that if the machine is in state $s$ and receives the input $i$, it will perform the output $o$ and reach the state $s'$ after $t$ time units. Similarly, if we consider that time is defined in stochastic terms, a transition as $s \xrightarrow{i/o}_\xi s'$ indicates that if the machine is in state $s$ and receives the input $i$, it will perform the output $o$ and reach the state $s'$ after a certain time $t$ with probability $F_\xi(t)$, where $F_\xi$ is the probability distribution function associated with $\xi$. Finally, in a model where non-determinism is probabilistically quantified and time is defined in stochastic terms, a transition such as $s \xrightarrow{i/o}_{p,\xi} s'$ indicates that if the machine is in state $s$ and receives the input $i$ then with probability $p$ the machine emits the output $o$ and it moves to state $s'$ before time $t$ with probability $F_\xi(t)$.

If time is expressed in stochastic terms, then the black-box assumption complicates the work of the tester. In this case, testers cannot compare in a direct way timed requirements of the *real* implementation with those established in the model. The idea is that we can *see* the random variable associated with a given transition in the model, but we cannot do the same with the corresponding transition of the implementation, since we do not have access to it. Thus, in contrast with approaches considering fix time values, to perform a transition of the implementation once does not allow the tester to obtain all the information about its temporal behavior. Therefore, the tester must induce the system to perform the same transition several times to collect different time values.

*Implementation relations* are used to relate systems under test and specifications. It is very helpful to start by considering an implementation relation where time is not taken into account. In this case, the idea is that the implementation $I$ does not *invent* anything for those inputs that are *specified* in the model. In order to cope with time, we do not take into account only that a system may perform a given action but we also record the amount of time that the system needs to do so. Several timed conformance relations can be defined according to the interpretation of *good* implementation and the different time domains. Time aspects add extra complexity to the task of defining these relations. For example, even though an implementation $I$ had the same traces as a formal model $S$, we should not consider that $I$ conforms to $S$ if $I$ is always *slower* than $S$. Moreover, it can be the case that a system performs the same sequence of actions for different times. These facts motivate the definition of several conformance relations. For example, it can be said that an implementation conforms to a formal model if the implementation is always *faster*, or if the implementation is at least as *fast* as the worst case of the model.

With respect to the application of tests to implementations, the above mentioned non-deterministic temporal behavior requires that tests work in a specific

manner. For example, if we apply a test and we observe that the implementation takes less time than the one required by the formal model, then this single application of the test allows us to know that the implementation *may* be faster than the model, but not that it *must* be so.

## 3 Other work on testing of timed and probabilistic systems

In addition to the work presented in this talk, I have also participated in the development of other frameworks to test timed and probabilistic systems that I would like to briefly review. First, it is worth mentioning that the presented framework is general enough so that it can be easily modified to deal with other formalisms such as timed variants of stream X-machines [14]. The timed framework presented in this talk allows their users to express temporal requirements concerning the time elapsed between the reception of an input and the production of an output but it does not deal with *timeouts*. Therefore, this work [16] was extended to add timeouts: if after a certain amount of time the system does not receive and input, then a timeout will be invoked and the system will change its state [15]. Another interesting line of work consists in considering that the time information might not exactly reflect reality (e.g. due to bad equipment to measure time). Therefore, the tester might assume that small errors can be acceptable [17].

The previous approaches consider *active* testing, that is, the tester provides inputs to the system under test and analyzes the received outputs. However, in certain circumstances the tester cannot interact with the system and has to analyze observations of this system that are not controlled by him: he becomes a *passive* tester. During the last years, I was interested on the definition of a passive testing methodology for timed systems [3] and in the application of the methodology to real systems [2, 1].

A different line of research in testing of timed systems consists in using genetic algorithms to select *better* test cases among the (possibly infinite) test cases that can be derived from a given specification [4, 5].

Concerning testing of probabilistic systems, it is interesting to test the probabilistic behavior of systems with distributed ports since it is challenging to establish probabilistic relations between ports and between actions in the same port [10].

In some situations it is difficult to precisely specify the probabilities governing the behavior of the system. Therefore, instead of specifying that a certain event will happen with probability $p$, it is more appropriate to say that it will happen with a probability belonging to the interval $[p - \epsilon, p + \delta]$. The definition of a testing methodology for this kind of systems was quite interesting [12].

The previous approaches considered extensions of the **ioco** implementation relation [22]. A more theoretical line of work consists in defining timed and probabilistic extensions [19, 20, 11] of the classical de Nicola & Hennessy testing framework [18, 8].

## 4 Future work

I continue working on probabilistic and/or timed extensions of formal testing frameworks. Currently, I am very interested on testing systems with distributed ports. One line of work consists in using (probabilistic and distributed) schedulers to solve some of the problems detected on previous work [10]. Another line of work considers the analysis of time while testing systems with distributed ports. Using this additional information can help to fix the order in which events at different ports were performed. I am also working on applying our frameworks to test real systems. Specifically, the timed framework presented in this talk [16] is being use to test the temporal behavior of major household appliances while the probabilistic and timed framework presented during the talk [13] is being applied to analyze wireless communication protocols.

## Acknowledgements

First, I would like to thank Burkhart Wolff and Fatiha Zaïdi, Chairs of the conference, for their kind invitation to be an invited speaker in ICTSS 2011. I would also like to thank all my colleagues with whom I have had the privilege of working on the study of timed and probabilistic aspects: César Andrés, Ana Cavalli, Karnig Derderian, David de Frutos, Carlos Gregorio, Rob Hierons, Iksoon Hwang, Luis Llana, Natalia López, Stéphane Maag, Mercedes G. Merayo, Pedro Palao, Fernando L. Pelayo, Ismael Rodríguez and Fernando Rubio.

## References

1. Andrés, C., Maag, S., Cavalli, A., Merayo, M., Núñez, M.: Analysis of the OLSR protocol by using formal passive testing. In: 16th Asia-Pacific Software Engineering Conference, APSEC'09. pp. 152–159. IEEE Computer Society (2009)
2. Andrés, C., Merayo, M., Núñez, M.: Applying formal passive testing to study temporal properties of the stream control transmission protocol. In: 7th IEEE Int. Conf. on Software Engineering and Formal Methods, SEFM'09. pp. 73–82. IEEE Computer Society (2009)
3. Andrés, C., Merayo, M., Núñez, M.: Formal passive testing of timed systems: Theory and tools. Software Testing, Verification and Reliability (2012), accepted for publication
4. Derderian, K., Merayo, M., Hierons, R., Núñez, M.: Aiding test case generation in temporally constrained state based systems using genetic algorithms. In: 10th Int. Conf. on Artificial Neural Networks, IWANN'09, LNCS 5517. pp. 327–334. Springer (2009)
5. Derderian, K., Merayo, M., Hierons, R., Núñez, M.: A case study on the use of genetic algorithms to generate test cases for temporal systems. In: 11th Int. Conf. on Artificial Neural Networks, IWANN'11, LNCS 6692. pp. 396–403. Springer (2011)
6. Gaudel, M.C.: Testing can be formal, too! In: 6th Int. Joint Conf. CAAP/FASE, Theory and Practice of Software Development, TAPSOFT'95, LNCS 915. pp. 82–96. Springer (1995)

7. Glabbeek, R.v., Smolka, S., Steffen, B.: Reactive, generative and stratified models of probabilistic processes. Information and Computation 121(1), 59–80 (1995)
8. Hennessy, M.: Algebraic Theory of Processes. MIT Press (1988)
9. Hierons, R., Bogdanov, K., Bowen, J., Cleaveland, R., Derrick, J., Dick, J., Gheorghe, M., Harman, M., Kapoor, K., Krause, P., Luettgen, G., Simons, A., Vilkomir, S., Woodward, M., Zedan, H.: Using formal methods to support testing. ACM Computing Surveys 41(2) (2009)
10. Hierons, R., Núñez, M.: Testing probabilistic distributed systems. In: IFIP 30th Int. Conf. on Formal Techniques for Distributed Systems, FMOODS/FORTE'10, LNCS 6117. pp. 63–77. Springer (2010)
11. Llana, L., Núñez, M.: Testing semantics for RTPA. Fundamenta Informaticae 90(3), 305–335 (2009)
12. López, N., Núñez, M., Rodríguez, I.: Specification, testing and implementation relations for symbolic-probabilistic systems. Theoretical Computer Science 353(1–3), 228–248 (2006)
13. Merayo, M., Hwang, I., Núñez, M., Cavalli, A.: A statistical approach to test stochastic and probabilistic systems. In: 11th Int. Conf. on Formal Engineering Methods, ICFEM'09, LNCS 5885. pp. 186–205. Springer (2009)
14. Merayo, M., Núñez, M., Hierons, R.: Testing timed systems modeled by stream X-machines. Software and Systems Modeling 10(2), 201–217 (2011)
15. Merayo, M., Núñez, M., Rodríguez, I.: Extending EFSMs to specify and test timed systems with action durations and timeouts. IEEE Transactions on Computers 57(6), 835–848 (2008)
16. Merayo, M., Núñez, M., Rodríguez, I.: Formal testing from timed finite state machines. Computer Networks 52(2), 432–460 (2008)
17. Merayo, M., Núñez, M., Rodríguez, I.: A formal framework to test soft and hard deadlines in timed systems. Software Testing, Verification and Reliability (2011), accepted for publication, doi: 10.1002/stvr.448
18. de Nicola, R., Hennessy, M.: Testing equivalences for processes. Theoretical Computer Science 34, 83–133 (1984)
19. Núñez, M.: Algebraic theory of probabilistic processes. Journal of Logic and Algebraic Programming 56(1–2), 117–177 (2003)
20. Núñez, M., Llana, L.: A hierarchy of equivalences for probabilistic processes. In: 28th IFIP Int. Conf. on Formal Techniques for Networked and Distributed Systems, FORTE'08, LNCS 5048. pp. 267–282. Springer (2008)
21. Sidhu, D., Leung, T.K.: Formal methods for protocol testing: A detailed study. IEEE Transactions on Software Engineering 15(4), 413–426 (1989)
22. Tretmans, J.: Model based testing with labelled transition systems. In: Formal Methods and Testing, LNCS 4949. pp. 1–38. Springer (2008)