26

Some Issues on Testing Theory and its Applications

Ana Cavalli[a] and Marc Phalippou [b]

[a] France Télécom - Institut National des Télécommunications, 9 rue Charles Fourier, 91011 Evry Cedex, France
E-mail: Ana.Cavalli@int-evry.fr

[b] France Télécom-CNET LAA/SLC/EVP, Route de Trégastel, BP 40, F-22301 Lannion Cedex, France, E-mail: Marc.Phalippou@lannion.cnet.fr

Abstract
This paper presents a summary of some issues that appear to be relevant to testing theory and its applications. These issues are linked to formal methods used for test derivation, to the semantic models chosen to represent these specifications, and to the appropriate techniques needed for selecting a reasonable test effort. A discussion is presented in a polemical fashion trying to illustrate the problems related to the translation of one model into another, to non deterministic and partially specified systems, to the size of the specifications and test suites and the ways to reduce them.

Keywords Codes : D.2.5; F.3.1; I.6.4
Keywords: Conformance Testing, Formal Description Techniques, Testing Theory.

1. INTRODUCTION

During the last years, testing theory for protocols (or more generally for telecommunication systems) has shown important progress in several directions, as can be seen from the number of papers presented in conferences like IWPTS, PSTV or FORTE. But the debate about real applications of this theory is still open. Some people claim that theory is not always applicable, because it is too far from giving the expected solution: efficiently testing of real telecommunication systems at a reasonable cost. The fact is that many issues of testing theory remain open. Researchers who develop the theory are investigating several approaches, which all have some strengths and some weaknesses. In this paper some of these open questions will be reviewed.

T. Mizuno et al. (eds.), *Protocol Test Systems*

Formal Description Techniques (FDT) have been recognized as useful methods for the specification of complex systems, such as telecommunications systems. In the field of protocol communication, the more widely spread FDTs are those based on process algebras, such as CCS, CSP and LOTOS and those based on Extended Finite State Machines, such as the SDL and ESTELLE. The semantic models for these two kinds of FDTs are, for the first ones, the Labelled Transition Systems (LTSs) and for the second ones, the Input/Output Finite State Machines (I/OFSMs), also known as Mealy machines.
All these formalisms have been used to study formal methods for conformance testing. Two classes of methods can be considered:

1. those based on the formal description of tests in the form of Input/Output sequences obtained from an I/OFSM. The best known methods are the Transition Tour (TT), the Distinction Sequence (DS), the W method, the Unique Input/Output sequence (UIO sequence) and their more recent improvements: UIOv, Wp and HSI [1] methods.

2. those based on the development through formal descriptions in the form of a process algebra, of a test process called canonical tester. These methods have been essentially applied to the LOTOS language, the best known being those based on the concept of implementation relation *conf* and its variants [2] [3] [4]. It can also be mentioned an adaptation of this approach to finite state machines [5].
However, all these methods raise a number of questions, as discussed in the following sections.

2. LABELLED TRANSITION SYSTEMS VERSUS I/OFSM

The first open question is to determine if one of these semantic models is more appropriate than the other one in order to study test questions. The most important characteristic of these models, i.e. their major difference, is the communication model which is used : communication by means of inputs and outputs in the case of I/OFSM, and communication by means of negotiated rendez-vous in the case of LTS. These different communication mechanisms explain the different results which have been obtained with the two models. Let us remind theses main results:

1. Basic concepts of I/OFSMs have been defined in the work of [6], [7], [8]. The methods for test derivation from I/OFSMs have their origin in the checking experiments from automata theory, in which the objective is to determine experimentally whether a given state table describes the behavior of a FSM implementation. All this works started with the paper of Moore.
Most of the derived methods (Distinguishing Sequences, Unique Input/Output Sequence and W method) grant a complete fault coverage with respect to a given type of faults (transfer and output faults), as long as it concerns a single completely specified, minimal and deterministic I/OFSM, and provided that the tester knows an upper bound on the number of states in the I/OFSM implementation. The usual assumption made in the

literature is that the implementation has no more states that the specification. Detected faults do not increase the number of states. It will be discussed in the next sections how the restrictions on models and implementations can be weakened.

2. The Labelled Transition Systems allow to model the specifications formulated in CCS, CSP or LOTOS. They are, in general, partially specified and non deterministic. For these systems, unspecified interactions produce deadlocks.
A big effort was made in the LOTOS community to define a theoretical frame for conformance testing. This work [2], [9] resulted in the definition of concepts such as canonical testers and conformance implementation relations. These theories focus on the fact that specifications may not be deterministic. The non determinism may be due to interactions with the environment as well as to internal actions of the system.
However, even if in some works algorithms are given to calculate canonical testers, the practical application of these concepts presents some difficulties: for instance, the fact that the test of an implementation is represented as the parallel composition of the canonical tester (which is modelled by a LOTOS process) and the implementation under test (which also is modelled by a LOTOS process). The test verdict is based on the existence of deadlocks in the system composed by the tester and the implementation. If no deadlock can occur in any parallel run of the tester and the implementation under test, then it will be concluded to a success and it can be affirmed that the implementation conforms to the specification. Therefore the success is defined as a global property of the test system. The test experiment is infinite if the specification has an infinite behavior.

As it can be seen the two different semantic models have initially lead to the development of very different theories. However, current research try to unify these works around some common fundamental concepts. Implementation relations have been introduced in the I/OFSM world, such as quasi-equivalence [1]. Canonical testers have also been defined for such model [10]. On the other hand, adaptation of I/OFSM based test generation methods to LTS has been studied (see for instance the UE method presented in [11]). This adaptation requires similar assumptions as in the case of I/OFSM, i.e. a known bound on the number of states of the implementation.

In order to establish some equivalence results between these theories, it has been proposed to translate one model into the other:

1. Translation of I/OFSMs into LTSs has been proposed in the context of the SPECS project [12]. In this project, which had as one of its aims the definition of a semantic model common to SDL and LOTOS, it has been proposed to translate SDL into LTSs. It seemed natural to translate the queues associated to processes in terms of an associated buffer process. These works were followed by others on asynchronous testing or testing in the context of queues [3]. They are attempts to develop models for SDL processes and their queues in terms of LTSs and to formulate a theory on testing taking this context into account. It should also be noted that the formal semantics of SDL proposed in the standard, is based on the translation of SDL specifications into LTSs [13].

2. In [10] it is proposed to translate LTSs into I/OFSMs. The FSM are constructed preserving the failure trace equivalence between LTSs. A similar proposal is done also in

[14], where LTSs are transformed in completely specified FSMs, applying the following principles: there is a state in the I/OFSM for each state in the LTS. Rejected actions are transformed in a looping transition in the corresponding state with the input corresponding to the rejected action and having as output the label *reject* and accepted actions are transformed in a transition from the corresponding state with the input corresponding to the accepted action and having as output *accept*. This transformation has the advantage of allowing the user to detect every error in the implementation and not to stop the testing procedure at the first error detected (as is the case in the classical testing theory for LOTOS).

Therefore, it could be concluded that there exists a basis for unifying the theories based on the different semantic models: implementation relations and testers can be adapted from one world to the other one. A last open issue is the final translation of testers into some recognized test notation, such as TTCN: since TTCN has an input-output based communication mechanism, translation from LTS into TTCN present some difficulties. Research is going on this subject.
Apart from the choice of a particular communication mechanism, strong debates about the characteristics of the models occurred in the test theory community. In the next two sections, two important issues which have been recently discussed are presented.

3. NON DETERMINISTIC VERSUS DETERMINISTIC SPECIFICATIONS

Some years ago, it was generally admitted that protocols (or telecommunication systems) should be modelled as deterministic automata. At that time, the only existing test theories were those based on I/OFSM, and all of them (W, DS, UIO) were based on deterministic models. But with the emergence of LTS based test theories, studies about the effect of non determinism (and the way to handle it in test situations) became a key issue.
Since "non determinism" is a word with many interpretations, let us first remind what kinds of non determinism are related to I/OFSM or LTS. An automaton is said to be non deterministic if:

1. for the same input, there are similar outputs and different next states (this is called classical automata non determinism);

2. for the same input, there are different outputs and each one ends in a different state (this is called observable test non determinism);

3. spontaneous transitions can occur in concurrence with observable labelled transitions. In the world of I/OFSM a spontaneous transition is modelled by a transition with a null input.
In the world of LTS a spontaneous transition is labelled by the special internal event τ. In the case of I/OFSM this non determinism is said to be observable if the outputs are different (it is observable test non determinism). In the case of LTS it is called LTS non determinism.

Solutions have been proposed to deal with these kind of non-determinism. Non observable non determinism is not detectable by means of testing (as indicated by its name). Observable non determinism is handled by improved versions of the classical test methods in the world of I/OFSM [1], [15], and by adaptation of these methods to LTS [16].

However, in any case, it is possible to obtain finite test suites in the case of non deterministic implementations only if some fairness hypothesis is made: "a certain fairness among the different behaviors allowed by the non determinism of the tested systems". This fairness assumption has been formalized in [5] and named "bounded fairness hypothesis".
In parallel with this progress of test theory, which allows now to cope with non determinism, it became widely accepted among the researchers that realistic test situations must be modelled using non deterministic models. The main reason is that even if the systems to be tested are intrinsically deterministic, the test can be performed only through a test interface [17], which limits the control that the tester has over the implementation under test. In this case, implementations appear as if they were non deterministic [10].

4. COMPLETELY SPECIFIED VERSUS PARTIALLY SPECIFIED SYSTEMS

In a similar way, some years ago, the only existing test theories (those based on I/OFSM) provided solutions to the test generation problem for completely specified automata. Let us remind that an I/OFSM automaton is completely specified if in each state there exists at least one transition corresponding to each one of the input events. If it is not the case, we say that the protocol is partially specified: the output symbol of some states for some input symbols may not be specified.

Here again there was a debate among researchers to decide whether protocols (or telecommunication systems) should be modelled as partially specified automata.
It should be noted that for certain languages, for instance SDL which could have a semantics defined in terms of I/OFSMs, there exists implicitly a completeness. In SDL all inputs are accepted and there is not blockage by an unexpected input. The system remains in the same state and this situation may be formally represented by adding a loop to the state in question, labelled with the unexpected event and an output null. This is called by some authors completeness assumption. In this case a strong conformance relation can be defined [18]. This hypothesis of completely specified I/OFSM is necessary to insure that the I/OFSM is minimal, in order to define state identifications.

New methods have been proposed for state reduction of partially deterministic I/OFSM, based on the merging of the so called compatible states [8]. A state A is said to be compatible to another state B, iff A can accept all inputs accepted by B with the same outputs, and possibly others. Extensions of traditional test generation methods, which can guarantee fault coverage even for partially specified automata, have been published: the most general method is called HSI (Harmonized State Identifier) [1]. These methods are based on the notion of quasi-equivalent states defined by [19]. They propose a different

proof from classical methods, which require to find a minimal machine equivalent to the specification.

Here again, now that the theory has provided solutions for partially specified automata, researchers have found good reasons to explain that models of realistic test situations should be partially specified. The influence of the test interface has been raised as such a good reason [10].

In the case of LTSs, the proposal of an unexpected rendez-vous produce a blockage. In that sense, there is no concept of unspecified reception of event, and all LTS must be considered as completely specified.

5. ISOLATED TESTING VERSUS EMBEDDING TESTING

The methods for test derivation from FSMs (TT, DS, UIO, W method) are all based on the following assumption: that they test a single I/OFSM. This means that these methods only permit to test each component of the system in isolation. But there is at least two situations in which the composition of several components should be considered:

1. When some structure is known on the system under test. In this case, the system under test may be represented as a composition of several automata. The classical methods cannot use information about this structure: the global behaviour must be computed before application of the method. Other methods that are based on a static description of extended FSM present the same limitation.
On the contrary, some of the methods proposed for LTSs and the associated tools [11], [20] permit to do a kind of grey box testing. One can be acquainted with the structure of the system, disregard the parts non concerned with testing and focus on the components of the systems to be tested. This advantage is linked to features of the languages based on the process algebra, which offer the possibility of specifying the system at different levels of abstraction. The advantage of this method is that the finite LTSs integrates at the same time the data and the control structure. In addition, the global system can be described and not only isolated processes as is the case for classical methods based on I/OFSMs. This possibility of abstraction offered by the LTSs has been as well applied to study interoperability testing [21], defined as the test of interactions among different implantations but also defined by certain authors as the test of the service of protocols.

2. When the system under test is tested through a test interface. This is the case for instance if the remote test architecture is used. In this case, the tester must be computed from the description of the specification composed with the test interface. Here again classical methods cannot use this composition. Some methods have been proposed to solve this point by means of approximations [22]. Research continues on this subject.

6. FINITE TESTERS VERSUS INFINITE TESTERS

As we mentioned above, in the theories for test derivation based on the LTS model, the implementation and the tester (in this case the canonical tester) may become infinite. This is due to the fact that the implementation relations which are checked by the testers are quite strong, and that no special assumption is made about the implementations under test.
In order to solve this problem, the concept of test hypothesis has been introduced [23]. A test hypothesis is an assumption which is made about the implementation under test. With this assumption, the test is limited to a reduced subset of all possible implementations, and we can easily understand that the test effort will be reduced since less properties have to be checked.
Such an hypothesis is made by all the classical test methods based on I/OFSM: it is assumed that an upper bound is known on the number of states of the implementation. Work has been done in order to adapt these methods to LTSs. These works represent a pragmatic approach to test generation from LOTOS specifications. Finite LTSs are considered and methods used for I/OFSMs are adapted, as for instance partial UIO sequences [14].

An open question is to find suitable test hypothesis which:
1. are likely to be verified by real implementations;
2. are powerful enough to reduce significantly the size of the testers.

In the protocol test community, the hypothesis on the upper bound of the number of states of implementations is the only one which is studied. One can wonder whether it is realistic. In effect, it is very difficult to assume a bound for the number of states of real communication devices, which may be incredibly complex, for instance switching systems.

On the other hand, in the classical software test community, a method called partition testing is often used [24]. It has been shown that this method corresponds to another kind of test hypothesis, called uniformity hypothesis in [23]. Application of such a method to protocol testing should maybe be more deeply investigated.

7. COMPLEXITY ISSUES

Several interesting results related to complexity issues have been presented in [25]. In this work, it is proved that to determine whether an I/OFSM has a *preset* distinguishing sequence (the input sequence is fixed ahead of time) is a PSPACE-complete problem. The same result is shown for UIO sequences : to determine if all states have an UIO sequence is also a PSPACE-complete problem. However, it is possible to find a polynomial time algorithm to determine if an I/OFSM has an *adaptative* distinguishing sequence. A sequence is *adaptative* if at each step of the test, the next input symbol depends on the observed previous outputs. They propose also a randomized polynomial algorithm to

calculate a checking sequence that allows to distinguish an I/OFSM from all other machines with the same number of states.

Finally, another important issue concerns the size of real specifications. Even in the case where the test methods define simple algorithms to be applied on the reachability graph of the systems, application to real systems is limited by the fact that it is usually not possible to compute this model. On this domain problems are no more test theory problems, but rather algorithmic problems. However, these problems are really key issues in the development of automatic test generation methods.
Some attempts were made to reduce the size, by applying, for instance, methods that allow a partial development of the model. These methods may be useful for developing test cases in function of given test purposes but without insuring any fault coverage.

REFERENCES

1. G. Luo, A. Petrenko and G.v. Bochmann, Selecting Test Sequences for Partially-Specified Nondeterministic Finite State Machines, Publication n. 864, Université de Montreal, Février 1993.
2. E. Brinksma, et al, A Formal Approach to Conformance Testing, IWPTS'91, Leidschendam Netherlands, October 1991.
3. Jan Tretmans and Louis Verhaard, A Queue Model Relating Synchronous and Asynchronous Communication, PSTV XII, 1992, Orlando, USA.
4. K. Drira, The Refusal Graph : a Tradeoff between Verification and Test, IFIP Transactions, Protocol Test Systems, VI, (the Proceedings of IFIP TC6 Fifth International Workshop on Protocol Test Systems, 1993), Ed. by O. Rafiq, 1994, North-Holland, pp.331-345.
5. M. Phalippou, Executable Testers, IFIP Transactions, Protocol Test Systems, VI, (the Proceedings of IFIP TC6 Fifth International Workshop on Protocol Test Systems, 1993), Ed. by O. Rafiq, 1994, North-Holland, pp.331-345.
6. Moore E. F., Gedanken-experiments on Sequential Machines, in Automata Studies, Annals of Mathematics studies, No 34, pp. 129-153, Princeton University Press, Princeton, USA.
7. F.C. Hennie, Fault-detecting experiments for sequential circuits, in Proceedings 5th Ann. Symp. on Switching Circuit Theory and Logical Design, November, 1964.
8. Z. Kohavi, Switching and Finite Automata Theory, McGraw-Hill, 1978.
9. Ed Brinksma, A theory for the Derivation of Tests, in S. Aggarwal (eds) Protocol Specification, Testing and Verification, VIII (North-Holland, Amsterdam, 1988)
10. A. Petrenko, G. v. Bochmann and R. Dssouli, Conformance relations and test derivation, IWPTS'93, Pau, France, September 1993.
11. A. Cavalli, S. Kim and P. Maigron, Automated Protocol Conformance Test Generation Based on Formal Methods for LOTOS Specifications, Proceedings of the 5th International Workshop on Protocol Test Systems, Montreal, September 1992.
12. Specification Environment for Communication Software, Project RACE R1046 (SPECS), 1988.

13. CCITT Blue Book Volume X - Fascile X.1 Functional Specification and Description Language (SDL) Recommendation Z.100 and Annexes A, B, C and E IXth Plenary Assembly, Melbourne, 14-25 November 1988.
14. A. Cavalli, S.U. Kim and P. Maigron, Improving Conformance Testing for LOTOS, FORTE'93, Boston, USA.
15. G. Luo, G. v Bochmann, A. Das and C. Wu, Failure-equivalent transformation of transition systems to avoid internal actions, Information Processing Letters 44, pp 333-343, 1992, North Holland.
16. S. Fujiwara and G. v. Bochmann, Testing Non-deterministic State Machines with Fault Coverage, IWPTS'91, Leidschendam, Netherlands, October 1991.
17. ISO SC 21 P.54 / ITU TS SG 10 Q.8, Formal Methods in Conformance Testing, Working document - approved Geneva output, October 1993.
18. K. Sabnani and A. Dahbura, A Protocol Test Generation Procedure, Computer Networks and ISDN Systems, Vol. 15, No. 4, pp. 285-297, 1988.
19. A. Gill, Introduction to the Theory of Finite-State Machines, McGraw-Hill, New York, 1962.
20. H. Garavel, The Open CAESAR Reference Manual, Laboratoire de Génie Informatique, Institut IMAG, Grenoble, May 1992.
21. R. Castanet and O. Kone, Deriving Coordinated Testers for Interoperability, IFIP Transactions, Protocol Test Systems, VI, (the Proceedings of IFIP TC6 Fifth International Workshop on Protocol Test Systems, 1993), Ed. by O. Rafiq, 1994, North-Holland, pp.331-345.
22. M. Phalippou, Relations d'implantation et hypotheses de test sur des automates a entrees et sorties, These de l'Universite de Bordeaux I, to be published in 1994.
23. M.C. Gaudel, Test Selection Based on ADT Specifications, Proceedings of the 5th International Workshop on Protocol Test Systems, Montreal, September 1992.
24. E. Weyuker, B. Jeng, Analysing Partition Testing Strategies, IEEE Transactions on Software Engineering, vol. 17, n. 7, July 1991.
25. M. Yannakakis and D. Lee, Testing Finite State Machines, Proceedings of 23rd Annual ACM Symposium on Theory of Computing, STOC 91, 1991.