Overcoming observability problems in distributed test architectures

J. Chen [a] R. M. Hierons [b] H. Ural [c]

[a] *School of Computer Science, University of Windsor, Windsor, Ontario, Canada*

[b] *School of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, United Kingdom*

[c] *School of Information Technology and Engineering, University of Ottawa, Ottawa, Ontario, Canada*

Key words: finite state machine, testing, observability, controllability

1 Introduction

In distributed testing, a distributed test architecture is used where a tester is placed at each port of the system under test (SUT) N and an input sequence is applied. When N is a state based system specified as a finite state machine (FSM) M an input sequence to be applied to N can be constructed from M; the input sequence is then called a test sequence or a checking sequence. The application of a test/checking sequence [5] in the distributed test architecture introduces the possibility of controllability and observability problems. These problems occur if a tester cannot determine either when to apply a particular input to N, or whether a particular output from N has been generated in response to a specific input, respectively [6].

For some specifications there does not exist an input sequence in which the testers can coordinate solely via their interactions with N [2,8]. In this case it is necessary for the testers to exchange external coordination messages over a dedicated channel during the application of the input sequence. Similarly, such coordination messages can be used to overcome observability problems [2,7]. However, sometimes we want to avoid the use of coordination messages since they require us to set up an additional communications network and this makes testing more expensive. In addition, coordination messages introduce delays and these delays can cause problems if we have timing issues in our testing. Let us suppose, for example, that in testing we wish to follow the input of x_1 at port p_1 with the input of x_2 at port p_2 ($p_1 \neq p_2$) and in order to achieve this we sent a coordination message from the tester at p_1 to the tester

at p_2 after x_1 has been input. If we require that the time between x_1 and x_2 being sent is at most t and the process of sending coordination messages takes time $t' > t$ then this approach is not appropriate. The timing issues can be particularly problematic if the SUT responds rapidly to inputs, relative to the network used for coordination messages [1] . See [4] for a discussion of some of the timing issues that arise in using coordination messages.

This paper investigates conditions that must be satisfied by an FSM for the existence of input sequences that can be applied in a distributed test architecture without encountering controllability and observability problems and without using external coordination messages. Such conditions have two potential values. First, they can be used to determine whether we require coordination messages and thus a network that connects the testers. Second, if we wish to avoid the use of coordination messages in testing then these conditions can be seen as testability conditions that can inform the design process. Results given in this paper differ from those in [3] in the following ways. First, the conditions are strictly weaker than those in [3] since we are less restrictive in the ways we achieve our goals. Second, [3] only considered observability problems; we consider both controllability and observability problems. In addition, [3] only considered a particular type of observability problem and we generalize this. Finally, we investigate the situation in which we need only add input sequences to complement a given test/checking sequence ρ and prove that the conditions for this problem are equivalent to those for the original problem.

2 Preliminaries

An *n-port Finite State Machine* M (simply called an FSM M) is defined as $M = (S, I, O, \delta, \lambda, s_0)$ where S is a finite set of states; $s_0 \in S$ is the initial state; $I = \bigcup_{i=1}^{n} I_i$, where I_i is the input alphabet of port i, and $I_i \cap I_j = \emptyset$ for $i, j \in [1, n]$, $i \neq j$; $O = \prod_{i=1}^{n}(O_i \cup \{-\})$, where O_i is the output alphabet of port i, and $-$ means null output; $\delta : S \times I \rightarrow S$ is the transition function; and $\lambda : S \times I \rightarrow O$ is the output function. Each $y \in O$ is a *vector of outputs* $\langle o_1, o_2, ..., o_n \rangle$ where $o_i \in O_i \cup \{-\}$ for $i \in [1, n]$. We use $*$ to denote any possible output, including $-$, at a port. We also use $*$ to denote any possible input or any possible vector of outputs. In the following, $p \in [1, n]$ is a port, $x \in I$ is a general input, and $x_p \in I_p$ is an input at p. We use $y \mid_p$ to denote the output at p in y. A *transition* of M is a triple $t = (s_1, s_2, x/y)$, where $s_1, s_2 \in S$, $x \in I$, and $y \in O$ such that $\delta(s_1, x) = s_2$, $\lambda(s_1, x) = y$. s_1 and s_2 are called the *starting state* and the *ending state* of t respectively. The *input/output pair* x/y is the *label* of t. T denotes the set of all transitions in

[1] Naturally, we might introduce a faster network for use in sending the coordination messages but this can further increase the cost of testing.

M.

A *path* $\rho = t_1\ t_2\ \ldots\ t_k$ $(k \geq 0)$ is a finite sequence of transitions such that for $k \geq 2$, the ending state of t_i is the starting state of t_{i+1} for all $i \in [1, k-1]$. When the ending state of the last transition of path ρ_1 is the starting state of the first transition of path ρ_2, we use $\rho_1\rho_2$ to denote the *concatenation* of ρ_1 and ρ_2. The *label* of a path $(s_1, s_2, x_1/y_1)\ (s_2, s_3, x_2/y_2)\ \ldots\ (s_k, s_{k+1}, x_k/y_k)$ $(k \geq 1)$ is the sequence of input/output pairs $x_1/y_1\ x_2/y_2 \ldots x_k/y_k$ which is an *input/output sequence*. The *input portion* of a path $(s_1, s_2, x_1/y_1)\ (s_2, s_3, x_2/y_2)$ $\ldots\ (s_k, s_{k+1}, x_k/y_k)$ $(k \geq 1)$ is the input sequence $x_1x_2 \ldots x_k$.

Given an FSM M and a sequence tt' of consecutive transitions, $t = (s_1, s_2, x/y)$ and $t' = (s_2, s_3, x'/y')$, a *controllability problem* occurs if the port p at which x' is input is not involved in t: $x \notin X_p$ and $y\ |_p{=}\ -$. If this problem occurs then the tester at p does not know when to send x' and so tt' cannot be applied in testing. Consecutive transitions t and t' form a *synchronizable pair* of transitions if t' can follow t without causing a controllability problem. A path in which every pair of transitions is synchronizable is called a *synchronizable path*. An input/output sequence is *synchronizable* if it is the label of a synchronizable path. We assume that for every pair of transitions (t, t') there is a synchronizable path that starts with t and ends with t'. If this condition does not hold, then the FSM is called *intrinsically non-synchronizable* and we cannot expect to be able to overcome the controllability problem [1]. A *same-port-output-cycle* in an FSM is a synchronizable path $(s_1, s_2, x_1/y_1)$ $(s_2, s_3, x_2/y_2)\ \ldots\ (s_k, s_{k+1}, x_k/y_k)$ $(k \geq 2)$ such that $s_1 = s_{k+1}$, $s_i \neq s_{i+1}$ for $i \in [1, k]$, and there exists a port p with $y_i\ |_p{\neq}\ -$ and $x_i \notin I_p$ for all $i \in [1, k]$. If such a cycle exists then there is no bound on the number of outputs the tester at port p can see without providing an input, a situation not too dissimilar to a livelock. We assume that any FSM considered is not intrinsically non-synchronizable and has no same-port-output-cycles.

Suppose that we are given an FSM M and a synchronizable path $t_1 \ldots t_k$ of M with label $x_1/y_1x_2/y_2 \ldots x_k/y_k$. An *output shift fault* in an implementation N of M exists if one of the following holds for some $1 \leq i < j \leq k$:

(1) There exists $p \in [1, n]$ and $o \in O_p$ such that $y_i\ |_p{=}\ o$ in M, for all $i < l \leq j$ we have that $y_l\ |_p{=}\ -$ in M, for all $i \leq l < j$ we have that N produces output $-$ at p in response to x_l after $x_1 \ldots x_{l-1}$, and N produces output o at p in response to x_j after $x_1 \ldots x_{j-1}$. Here the output o shifts from being produced in response to x_i to being produced in response to x_j and the shift is *between* t_i and t_j.

(2) There exists $p \in [1, n]$ and $o \in O_p$ such that $y_j\ |_p{=}\ o$ in M, for all $i \leq l < j$ we have that $y_l\ |_p{=}\ -$ in M, for all $i < l \leq j$ we have that N produces output $-$ at p in response to x_l after $x_1 \ldots x_{l-1}$, and N produces output o at p in response to x_i after $x_1 \ldots x_{i-1}$. Here the output o shifts

from being produced in response to x_j to being produced in response to x_i and the shift is *between* t_j and t_i.

An instance of the observability problem manifests itself as a *potentially undetectable output shift fault* if there is an output shift fault related to $o \in O_p$ in two transitions with labels x_i/y_i and x_j/y_j, such that $x_{i+1} \dots x_j \notin I_p$. The tester at p will not be able to detect the faults since it will observe the expected sequence of interactions in response to $x_i \dots x_j$. Let $\mathcal{T}_p$ denote the transitions of M that can be involved in potentially undetectable output shift faults. Thus $t \in \mathcal{T}_p$ if there exists a transition t' and a synchronizable path $t\rho t'$ or $t'\rho t$ of M such that there is a potentially undetectable output shift fault between t and t'. We want a test/checking sequence that is free from observability problems. Note that [3] only considers observability problems in which the two transitions involved in the shift are adjacent and thus $j = i + 1$; these are called *1-shift output faults*.

3 Definitions of leading and trailing paths

To verify the output of a transition t at port p a test/checking sequence must contain t within a context that leads to its output at p being identified.

Definition 1 *Given transition $t = (s_1, s_2, x/y)$, a synchronizable path $\rho_1 t \rho_2$ is said to be a* verifying path *for (t, p) if the following holds: for every synchronizable path $\rho = \rho'_1 \rho_1 t \rho_2 \rho'_2$ of M with starting state s_0, if the tester at p sees the expected sequence of inputs and outputs when the input portion of ρ is applied to the SUT then we can deduce that when ρ was applied the SUT* must *have produced output $y|_p$ at p in response to the input of x after the input portion of $\rho'_1 \rho_1$. We call ρ_1 a* leading path *for (t, p), and ρ_2 a* trailing path *for (t, p). When (t, p) has a verifying path, we also say that (t, p) is* verifiable.

If we have a verifying path $\rho_1 t \rho_2$ for (t, p) then we can embed this within *any* test/checking sequence and we know that if no failure is observed when the test/checking sequence is applied to the SUT then the SUT must have produced the expected output at p in response to the input x that was intended to trigger t. This allows us to check the output of t at p but relies on us knowing that the corresponding transition of N is executed when expected. This is the case if either it is known that every transition of N has the required final state or if the final state of each transition is verified in another part of the test/checking sequence. This paper concerns the issue of overcoming observability problems and so we assume that the final state of each transition is either known to be correct or is verified through some other means. In this paper, we consider the existence of *absolute verifying paths* for (t, p) where t has non-empty output.

Definition 2 *Given transition $t = (s_1, s_2, x/y)$ where $y|_p \neq -$, ρ_1 is an* absolute leading path *for (t, p) if either $\rho_1 = \varepsilon$ and $x \in I_p$ or $\rho_1 \neq \varepsilon$ and: $\rho_1 t$ is a synchronizable path; all transitions in ρ_1 have non-empty output at port p; and the first transition, and only the first transition in ρ_1 has input at p. Path ρ_2 is an* absolute trailing path *for (t, p) if: $t\rho_2$ is a synchronizable path; all transitions in ρ_2, possibly except the last, have non-empty output at port p; and the last transition, and only the last transition in ρ_2 has input at p.*

No matter how $\rho = \rho_1 t \rho_2$ is concatenated with other sequences, we can determine the output sequence at p in response to the first $|\rho| - 1$ inputs of ρ as this is immediately preceded and followed by input at p. Further, since we expect $|\rho| - 1$ outputs at p within this output sequence, and there are $|\rho| - 1$ corresponding inputs, the output of t at p must have been correct if the correct sequence of observations was seen at p. Thus, absolute verifying paths are verifying paths. Note that the conditions ensure that ρ_1 and ρ_2 cannot be shortened without violating the required properties.

4 The goals

Recall that $\mathcal{T}_p$ denotes the set of transitions involved in potentially undetectable output shift faults at port p in M. If transition t has output y then $t|_p$ denotes $y|_p$. Let $\mathcal{T}'_p = \mathcal{T}_p \cap \{t \mid t|_p \neq -\}$ denote the set of transitions involved in potentially undetectable output shift faults at p whose output at p are non-empty. The first goal is to determine if (t, p) is verifiable for every $p \in [1, n]$ and $t \in \mathcal{T}_p$. If this is the case then we can produce a verifying path for each (t, p) and include these in a test or checking sequence to check the output of every transition of the SUT at every port without suffering from controllability or observability problems.

Let $\mathcal{T}_{\rho,p}$ denote the set of transitions involved in potentially undetectable 1-shift output fault at p in ρ: $t \in \mathcal{T}_{\rho,p}$ if there exists a transition t' such that tt' or $t't$ is a synchronizable path in which there is a potentially undetectable output shift fault at p. $\mathcal{T}'_{\rho,p} = \mathcal{T}_{\rho,p} \cap \{t \mid t|_p \neq -\}$ denotes the set of transitions that are involved in potentially undetectable 1-shift output faults at p in ρ and have non-empty output at p. The second goal is: given a test/checking sequence ρ, determine if (t, p) is verifiable for every p and t such that t is the first or last transition in ρ or $t \in \mathcal{T}_{\rho,p}$. This appears to weaken the requirements since we are simply verifying that there is no potentially undetectable 1-shift output faults within a given ρ or at the first/last transition.

Below, we present necessary and sufficient condition for (t, p) to have an absolute verifying path for every p and $t \in \mathcal{T}'_p$ and show that this achieves the first goal. Then, we prove that the condition is the same for the second goal.

Theorem 1 *Let M be a given FSM which is not intrinsically non-synchronizable and has no same-port-output-cycles. Let p be any port of M.*

(i) *(t_0, p) has an absolute leading path for every $t_0 \in \mathcal{T}'_p$, if and only if $\forall t = (s_1, s_2, x/y) \in \mathcal{T}'_p$, $x \notin I_p$ implies $\exists (s_3, s_1, x'/y') \in T$ synchonizable with t such that $y'|_p \neq -$;*

(ii) *(t_0, p) has an absolute trailing path for every $t_0 \in \mathcal{T}'_p$, if and only if $\forall t = (s_1, s_2, x/y) \in \mathcal{T}'_p$, $\exists (s_2, s_4, x'/y') \in T$ synchonizable with t such that $x' \in I_p \vee y' \mid_p \neq -$.*

Proof

We prove part (i); part (ii) follows in a similar way. ($\Leftarrow$) Consider some $t_0 \in \mathcal{T}'_p$; we prove that there is an absolute leading path σ_0. If the input of t_0 is at p, $\sigma_0 = \varepsilon$. Suppose that the input of t_0 is not at p. We use proof by contradiction: suppose t_0 has no absolute leading path and let σ denote a longest path such that σt_0 is synchronizable, every transition in σ has non-empty output at p and no transition in σ has input at p. Since M has no same-port-output-cycles and has a finite number of states there must exist such a (finite) σ. Let $t_2 = (r_3, r_4, x_2/y_2)$ be the first transition of σ and thus $x_2 \notin I_p$.

Suppose $t_2 \in \mathcal{T}'_p$. Since $x_2 \notin I_p$, according to the condition, there exists a transition $t_3 = (r_5, r_3, x_3/y_3)$, such that $t_3 t_2$ is synchronizable and $y_3 \mid_p \neq -$. Suppose instead that $t_2 \notin \mathcal{T}'_p$. Since M is not intrinsically non-synchronizable, there exists a transition $t_3 = (r_5, r_3, x_3/y_3)$ such that $t_3 t_2$ is synchronizable. As $t_2 \notin \mathcal{T}'_p$, we know that $y_3 \mid_p \neq -$. In each case, since t_0 has no absolute leading path, $x_3 \notin I_p$ and so by considering $t_3\sigma$ we contradict the maximality of σ as required.

($\Rightarrow$) Consider a transition $t = (r_1, r_2, x/y) \in \mathcal{T}'_p$ where $x \notin I_p$, $y \mid_p \neq -$. Let σ denote an absolute leading path for t. Since $x \notin I_p$, $\sigma \neq \varepsilon$. By definition, the last transition of σ must have non-empty output at p and must be synchronizable with t and so the result follows. □

We now consider the problem of checking the output of transition t at p where $t|_p = -$. We prove that if we can verify the output of every transition t at p such that $t|_p \neq -$ then we can verify the output of every transition at p.

Definition 3 *Let R be a set of transitions in M. The synchronizable path ρ is an* absolute verifying path for (t, p) upon R *if we know that the output of t at p must be correct whenever the following hold:*

(1) *The output at p of every transition in R is correct in the SUT N; and*

(2) *There exists a synchronizable path $\rho' \rho \rho''$ in M that starts at s_0 such that the tester at p sees the expected sequence of observations when the input*

portion of $\rho'\rho\rho''$ is applied to N.

This says that if we have an absolute verifying path ρ for (t, p) upon R and we know that the transitions in R are correct then we can use *any* synchronizable path that contains ρ in order to check the output of t at p. The following shows that if we can produce absolute verifying paths for each $t \in \mathcal{T}'_p$ then we can also check the output at p of any $t \notin \mathcal{T}'_p$.

Theorem 2 *Given any FSM M that is not intrinsically non-synchronizable and port p, every transition $t \notin \mathcal{T}'_p$ has an absolute verifying path upon $\mathcal{T}'_p$.*

Proof

Consider transition t with empty output at p. Find a synchronizable path $\rho = \rho_1 t_m = t_1 \dots t_m$ $(m \geq 2)$ in M such that $t = t_j$ for some $j \in [1, m-1]$ and both t_1 and t_m have input at p. The existence of such a path is guaranteed since M is not intrinsically non-synchronizable. Since t_1 and t_m have input at p, if we embed ρ within a path $\rho'\rho\rho''$ we can determine, from the observations at p, the output produced at p in response to the input portion of ρ_1. If the output of t' at p is correct for all $t' \in \mathcal{T}'_p$, then when the input portion of ρ is applied we know that the correct output is produced by every transition $t' \in \mathcal{T}'_p$ from ρ_1. Thus, if the expected number of outputs are observed at p when the input portion of ρ is applied then the output of t at p must be empty and so is correct. Thus ρ is an absolutely verifying path for (t, p) upon $\mathcal{T}'_p$. □

This allows us to use weaker hypotheses than in [3]: the result in [3] included conditions that deal with transitions in $\mathcal{T}_p \setminus \mathcal{T}'_p$. In addition, [3] does not consider the controllability problem and considered only 1-shift output faults.

The second goal concerns the problem of verifying the outputs of those transitions that could be involved in a potentially undetectable 1-shift output fault in a test/checking sequence ρ plus the first and last transitions [2] . We therefore assume that ρ contains every transition of M and prove that the conditions given above cannot be weakened. Observe that this problem was not considered in [3]. Again, we first consider pairs (t, p) such that $t|_p \neq -$.

Lemma 1 *Given an FSM M and a port p, let $t_1 t_2$ be a synchronizable transition sequence such that $t_1|_p \neq -$ and $t_2|_p \neq -$. Then*

- *(t_1, p) has an absolute leading path $\Rightarrow$ (t_2, p) has an absolute leading path.*
- *(t_2, p) has an absolute trailing path $\Rightarrow$ (t_1, p) has an absolute trailing path.*

Proof

[2] We include the first and last transitions of ρ since we will combine ρ with other sequences to form a single test/checking sequence

We prove the first part (the proof of the second part is similar). If the input of t_2 is at p, then ϵ is a leading path of (t_2, p). If the input of t_2 is not at p, since the outputs of t_1 and t_2 at p are non-empty, ρ is an absolute leading path of (t_1, p) implies ρt_1 is an absolute leading path of (t_2, p). □

Theorem 3 *Given FSM M, port p, and synchronizable test/checking sequence $\rho = t_1 \dots t_m$, if for every $t' \in \mathcal{T}'_{\rho,p} \cup \{t_1, t_m\}$ there is an absolute verifying path of (t', p), then there is an absolute verifying path of (t, p) for every $t \in \mathcal{T}'_p$.*

Proof

Consider the leading path only (the part for trailing paths is similar). Let t^* be any transition in ρ with non-empty output at p, where $t^* \notin \mathcal{T}'_{\rho,p} \cup \{t_1, t_m\}$. Let $\rho' = t'_1 \dots t'_k$ $(k \geq 2)$ be a subsequence of ρ such that $t'_k = t^*$; $t'_1 = t_1$ or t'_1 has empty output at p; and $\forall i \in [2, k-1]$, t'_i has non-empty output at p. Since $t^* \notin \mathcal{T}'_{\rho,p} \cup \{t_1, t_m\}$ we know there is such a subsequence with $k \geq 2$.

If t'_1 has empty output at p, since t'_2 has non-empty output at p, we know that t'_2 has an absolute leading path. This is because if the input of t'_2 is at p then ε can be used as an absolute leading path; if the input of t'_2 is not at p, then $t'_2 \in \mathcal{T}'_{\rho,p}$, so according to the condition t'_2 has an absolute leading path ρ'. If t'_1 has non-empty output at p, then $t'_1 = t_1$ and so t'_1 has an absolute leading path. Since both t'_1 and t'_2 have non-empty output at p, by Lemma 1, t'_2 has an absolute leading path ρ'. Thus, in both cases, t'_2 has an absolute leading path ρ'. Clearly, $\rho' t'_2 \dots t'_{k-1}$ is an absolute leading path for t^* as required. □

The proof of the following is equivalent to the proof of Theorem 2.

Theorem 4 *Given FSM M that is not intrinsically non-synchronizable and port p, for any transition t with empty output at p, (t, p) has a verifying path upon $\mathcal{T}'_{\rho,p}$.*

Thus, there exist sequences to find all potentially undetectable 1-shift output faults in a test/checking sequence ρ, that contains every transition of M, if and only if we can overcome all possible observability problems in M.

5 Conclusions

This paper investigated conditions that must be satisfied by a specification in order for us to be able to produce a test/checking sequence that is free from controllability and observability problems. This problem is represented in the following way. For each transition t and port p we wish to produce a path $\rho_1 t \rho_2$ that checks the output of t at p. The effectiveness of $\rho_1 t \rho_2$, at checking

the output of t at p, must not be affected by controllability and observability problems. This paper gives conditions for the existence of such a path for each transition t and port p for a class of FSMs. This class of FSMs is strictly larger than that considered in [3] and the conditions produced are strictly weaker than those given in [3]. Interestingly, we also proved that these conditions are not weakened if we only wish to find potentially undetectable 1-shift output faults in a given test/checking sequence.

6 Acknowledgements

This work was supported in part by Natural Sciences and Engineering Research Council (NSERC) of Canada under grant RGPIN 976 and 209774, Leverhulme Trust grant number F/00275/D, Testing State Based Systems, and Engineering and Physical Sciences Research Council grant number GR/R43150, Formal Methods and Testing (FORTEST).

References

[1] S. Boyd and H. Ural. The synchronization problem in protocol testing and its complexity. *Information Processing Letters*, 40(3):131–136, 1991.

[2] L. Cacciari and O. Rafiq. Controllability and observability in distributed testing. *Information and Software Technology*, 41:767–780, 1999.

[3] J. Chen, R. M. Hierons, and H. Ural. Conditions for resolving observability problems in distributed testing. In *24rd IFIP International Conference on Formal Techniques for Networked and Distributed Systems (FORTE 2004)*, volume 3235 of *LNCS*, pages 229–242. Springer-Verlag, 2004.

[4] A. Khoumsi. A temporal approach for testing distributed systems. *IEEE Transactions on Software Engineering*, 28(11):1085–1103, 2002.

[5] D. Lee and M. Yannakakis. Principles and methods of testing finite–state machines – a survey. *Proceedings of the IEEE*, 84(8):1089–1123, 1996.

[6] G. Luo, R. Dssouli, G. v. Bochmann, P. Venkataram, and A. Ghedamsi. Test generation with respect to distributed interfaces. *Computer Standards and Interfaces*, 16:119–132, 1994.

[7] O. Rafiq and L. Cacciari. Coordination algorithm for distributed testing. *The Journal of Supercomputing*, 24:203–211, 2003.

[8] K.-C. Tai and Y.-C. Young. Synchronizable test sequences of finite state machines. *Computer Networks and ISDN Systems*, 30(12):1111–1134, 1998.