Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Information Processing Letters 88 (2003) 133–141

Information Processing Letters

www.elsevier.com/locate/ipl

Distributed testing without encountering controllability and observability problems ☆

Hasan Ural *, David Whittier

School of Information Technology and Engineering, University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada

Received 6 March 2001; received in revised form 30 January 2003

Abstract

The objective of testing is to determine whether a system under test conforms to its specification. In distributed test architectures that utilize remote testers, this objective can be complicated by the fact that testers may encounter problems relating to controllability and observability during the application of a test sequence. Existing solutions to these problems involve first constructing a test sequence from the specification of an implementation under test, and then inserting coordination messages or appending selected test subsequences that prevent the occurrences of controllability and observability problems during the application of the resulting test sequence. This paper proposes a method that utilizes a set of transformation rules to construct an auxiliary directed graph from a given specification, and constructs a rural Chinese postman tour in this graph to yield a minimum-length test sequence where there is no potential controllability or observability problems, and where the use of coordination messages is minimized.

Keywords: Distributed testing; Distributed test architectures; Distributed systems; Controllability and observability problems; Finite state machine; Test sequence construction

1. Introduction

Finite state machine (FSM) model has been widely used to specify behavior of systems in diverse areas such as sequential circuits, lexical analysis, pattern matching, and more recently machine learning and telecommunications. A particular example for the use of the FSM model is the specification of the control structures of distributed systems which are event-driven systems whose functionality are well-characterized by state-based models in terms of sequences of interactions between the system and its environment [7,18]. Such a system has a number of interfaces, called *ports*, that are remote to each other. Programs or users at these ports are sources of inputs and destinations of outputs of the system.

When testing a distributed system to ensure its correct functionality, a *distributed test architecture* is needed where a tester is placed at each port of the *system under test* (SUT) and a *test sequence* (sequence of input/output pairs constructed from the FSM model of the SUT) is applied. During the application of a test sequence within a distributed test architecture, the existence of multiple testers brings out the possibility of coordination problems among testers known as *con-*

☆ This work is supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.
* Corresponding author.
E-mail address: ural@site.uottawa.ca (H. Ural).

0020-0190/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0020-0190(03)00364-8

trollability and *observability* problems. These problems occur if a tester cannot determine either when to apply a particular input to a SUT, or whether a particular output from a SUT is generated in response to a specific input, respectively.

Various FSM-based test sequence construction methods have been proposed in the literature. An excellent survey of these methods can be found in [13]. Since these methods disregard the existence of multiple ports of an FSM, the application of the resulting test sequences in a distributed test architecture requires test coordination procedures to coordinate the actions of remote testers. Examples of these procedures for testing communication protocols and open distributed processing systems can be found in [11] and [12], respectively. Test coordination procedures bring additional setup and execution costs and introduce delays that may be unacceptable in some cases such as testing real-time systems [6].

It has been observed that the test coordination procedures can be avoided either by eliminating the need for explicit coordination among remote testers or by allowing testers to communicate with each other directly by exchanging external coordination messages. In the first approach, each pair of consecutive input/output pairs in a test sequence is formed such that there will be no controllability problem [15] and additional sequences of input/output pairs are appended to the test sequence such that each observability problem is eliminated [10,14,21]. Clearly, this approach can only be applied to an FSM which has an *intrinsically synchronizable* test sequence, i.e., an FSM in which every transition can be followed by at least one transition without creating a controllability problem. An efficient algorithm to determine the existence of an intrinsically synchronizable test sequence for an FSM is given in [2] which also proves that finding a shortest such sequence is NP-complete. Based on this result, various methods for constructing minimum-length intrinsically synchronizable test sequences that are not necessarily of optimal length have been proposed [4,9,16,17,20]. When there is an intrinsically synchronizable test sequence for an FSM, the first approach is completed by appending a test subsequence, without creating a controllability problem, to the given test sequence to eliminate each observability problem identified within the sequence [14]. In the second approach, the external coordination message exchanges between remote testers relating to either controllability or observability are inserted wherever there is a controllability or observability problem identified within the given test sequence [3].

A common feature of the existing methods for the simultaneous solution of both controllability and observability problems is that they start with a given test sequence and examine this sequence for the instances of potential controllability and observability problems in an attempt to eliminate them in the given sequence [3,14]. There is no attempt made however to reduce the number of additional test subsequences to be appended in [14] or to reduce the number of external coordination messages to be exchanged in [3].

In contrast, the method proposed in this paper will not examine a given test sequence for the instances of potential controllability and observability problems. Instead, it will identify potential controllability and observability problems before constructing a test sequence from a given specification and construct a minimum-length test sequence that can be applied without encountering controllability and observability problems in a distributed test architecture. This is achieved as follows. If the given FSM has an intrinsically synchronizable test sequence, the method eliminates the need for explicit coordination between testers by constructing an intrinsically synchronizable test sequence where the number of additional test subsequences appended for eliminating observability problems is minimized. Otherwise, the method constructs a test sequence where the number of external coordination messages exchanged between testers for eliminating controllability and observability problems is minimized.

The rest of the paper is organized as follows: Section 2 reviews the preliminaries. Section 3 presents the proposed method and compares it with existing methods. Section 4 gives the conclusions.

2. Preliminaries

An *n-port Finite State Machine* (*np*-FSM) $M = (S, \Sigma, \Gamma, \delta, \lambda, s_0)$ where

– S is a finite set of states of M and $s_0 \in S$ is the initial state of M,
– Σ is an n-tuple $(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$, where Σ_k is the input alphabet of port k, and $\Sigma_i \cap \Sigma_j = \emptyset$ for

$i \neq j,\ i, j = 1, 2, \ldots, n$; $I = \Sigma_1 \cup \Sigma_2 \cup \cdots \cup \Sigma_n \cup \{–\}$, where – means null input,
- Γ is an n-tuple $(\Gamma_1, \Gamma_2, \ldots, \Gamma_n)$, where Γ_k is the output alphabet of port k, and $\Gamma_i \cap \Gamma_j = \emptyset$ for $i \neq j,\ i, j = 1, 2, \ldots, n$; $O = \{\langle a_1, a_2, \ldots, a_n\rangle \mid \exists a_i \in \Gamma_i \cup \{–\},\ 1 \leqslant i \leqslant n\}$, where – means null output,
- δ is the transition function that maps $S \times I$ to S, i.e., $\delta : S \times I \to S$, and
- λ is the output function that maps $S \times I$ to O, i.e., $\lambda : S \times I \to O$.

A *transition* of an np-FSM M is a triple $(s_j, s_k; x/y)$, where $s_j, s_k \in S$, $x \in I$, and $y \in O$ such that $\delta(s_j, x) = s_k$ and $\lambda(s_j, x) = y$. M is represented by a digraph $G = (V, E)$ where V represents the set S of states of M, and E represents all specified transitions of M. Thus, each edge $e_{jk} = (v_j, v_k; x/y) \in E$ represents a state transition from state s_j to state s_k with input x and output y where *label*(e_{jk}) is the *input/output pair* x/y.

In a distributed test architecture involving multiple testers, each tester executes a local test sequence constructed from the global test sequence of a given np-FSM M. In the local test sequence, tester k knows only that there are transitions involving itself or others, and does not know the inputs or outputs of the other testers. A *global test sequence* ω of an np-FSM M is the label of a path of the digraph G representing M that is of the form $x_1/y_1, x_2/y_2, \ldots, x_m/y_m$, where $x_i \in I$ and $y_i \in O,\ 1 \leqslant i \leqslant m$. Given a global test sequence ω, a *local test sequence* ω_k for port k of an np-FSM M is of the form $\alpha_1, \alpha_2, \ldots, \alpha_m$, where α_i $(1 \leqslant i \leqslant m)$ is either x_i/y_i, $x_i \in \Sigma_k \cup \{–\}$, $y_i = \langle a_1, a_2, \ldots, a_n\rangle$, where $a_k \in \Gamma_k \cup \{–\}$ and $a_j = –$, for $j \neq k,\ 1 \leqslant j,\ k \leqslant n$; or an external coordination message transmission or reception to ensure controllability or observability.

Controllability refers to the ease of affecting the specified outputs. A *controllability* (*synchronization*) *problem* exists when a tester is required to send an input to the SUT in the current transition, and because it is not *involved* in the previous transition, i.e., it did not send the input or receive the output in the previous transition, it does not know when to send the input. Formally, given an np-FSM M and a global test sequence $\omega = x_1/y_1, x_2/y_2, \ldots, x_m/y_m$ of M, where $x_i \in I$ and $y_i \in O,\ 1 \leqslant i \leqslant m$, a (*controllability*) *synchronization problem* occurs when, in the labels x_j/y_j and x_{j+1}/y_{j+1} of any two consecutive transitions, there exists a tester k that sends x_{j+1} that is neither the one sending x_j nor one of those receiving an output belonging to $y_j,\ 1 \leqslant j \leqslant m-1$. Let tester h be the tester that sends x_j. As it will be used in the proposed method, the solution to the synchronization problem is to insert an external coordination message exchange $\langle -C_k, +C_h\rangle$ relating to controllability between x_j/y_j and x_{j+1}/y_{j+1} in the global test sequence which will result in the insertion of:

- "$-C_k$" in the local test sequence of tester h just after tester h sends x_j to imply that tester h is to send an external coordination message to tester k, and
- "$+C_h$" in the local test sequence of tester k just before tester k sends x_{j+1} to imply that tester k is to receive an external coordination message from tester h notifying tester k to send its input to the SUT.

Given an np-FSM M and a global test sequence $\omega = x_1/y_1, x_2/y_2, \ldots, x_m/y_m$ of M, where $x_i \in I$ and $y_i \in O,\ 1 \leqslant i \leqslant m$, any two consecutive transitions t_{ij} and t_{jk} whose labels are x_j/y_j and x_{j+1}/y_{j+1} form a *synchronizable pair* of transitions if t_{jk} can follow t_{ij} without generating a synchronization problem. Any subsequence of transitions in which every pair of transitions is synchronizable is called a *synchronizable transition sequence*. A global test sequence is said to be *synchronizable* if it is the label of a synchronizable transition sequence.

Observability refers to the ease of determining if specified inputs affect the outputs. An *observability problem* exists when a tester is expecting to receive an output from the SUT in response to either the previous input or the current input and because it did not send the current input, it does not know when to start or stop waiting for the output. The observability problem manifests itself as an undetectable output shift fault. Formally, given an np-FSM M and a global test sequence $\omega = x_1/y_1, x_2/y_2, \ldots, x_m/y_m$ of M, where $x_i \in I$ and $y_i \in O,\ 1 \leqslant i \leqslant m$, a 1-*shift output fault* in an implementation N of M occurs when, in the labels x_j/y_j and x_{j+1}/y_{j+1} of any two consecutive transitions, there exists one a_k in $y_j = \langle a_1, a_2, \ldots, a_n\rangle$ of M that occurs in $y_{j+1} =$

$\langle a_1, a_2, \ldots, a_n\rangle$ of N and not in y_j of N; or there exists one a_k in $y_{j+1} = \langle a_1, a_2, \ldots, a_n\rangle$ of M that occurs in $y_j = \langle a_1, a_2, \ldots, a_n\rangle$ of N and not in y_{j+1} of N, $1 \leqslant j \leqslant m-1$, $1 \leqslant k \leqslant m$.

In general, if there is a 1-shift output fault related to $a_k \in \Gamma_k$ in any two consecutive transitions whose labels are x_j/y_j and x_{j+1}/y_{j+1}, this fault will be *undetectable* by tester k that satisfies the condition

$$(a_k \text{ is in } y_j \text{ of } M \text{ XOR } a_k \text{ is in } y_{j+1} \text{ of } M) \text{ AND } x_{j+1} \notin \Sigma_k.$$

In this case, tester k is said to be *involved* in the shift. Let tester h be the tester that sends x_{j+1}. As it will be used in the proposed method, the solution to the observability problem is to insert an external coordination message exchange $\langle -O_k, +O_h\rangle$ relating to observability between x_j/y_j and x_{j+1}/y_{j+1} in the global test sequence which will result in the insertion of:

Case 1: ($a_k \in y_{j+1}$ of M) an external coordination message "$+O_h$" in the local test sequence for tester k just before receiving a_k. Tester h will have a message "$-O_k$" in its local test sequence just before sending x_{j+1}. By this exchange, tester h informs tester k to expect receiving an output from the SUT.

Case 2: ($a_k \in y_j$ of M) an external coordination message "$+O_h$" in the local test sequence for tester k just after receiving a_k. Tester h will have a message "$-O_k$" in its local test sequence just before sending x_{j+1}. By this exchange, tester h informs tester k that it must have received an output from the SUT by now.

3. The proposed method

Formally, the proposed method is a solution of the following problem: consider a minimal and deterministic np-FSM M represented by a strongly connected digraph $G = (V, E)$. Let $\Phi(M)$ be the set of all those implementations of M, each of which has no transfer fault and has the same sets of states, inputs, outputs and the initial state as M. Construct a global test sequence ω that distinguishes M from any faulty implementation of M in $\Phi(M)$ such that ω is minimal in terms of the number of both transitions of M and external coordination messages, and that ω can be applied in a distributed test architecture without encountering any controllability or observability problems.

3.1. Description of the proposed method

In the first phase of the proposed method, a digraph $G' = (V', E')$ is constructed from $G = (V, E)$ of M such that in any traversal of G', any two adjacent edges of G that are covered by the traversal of G' will not create a potential controllability problem.

Step 1. For each vertex $v_i \in V$,
- create a set of vertices $i^1, i^2, \ldots, i^n$ in V', and
- create $\forall j, k,\ j \leqslant n,\ k \leqslant n,\ j \neq k$, a set of dashed edges $(i^j, i^k; \langle -C_k, +C_j\rangle)$ and $(i^k, i^j; \langle -C_j, +C_k\rangle)$ in E'.

Step 2. For each edge $e_{jk} = (v_j, v_k; x/y) \in E$, create the following edge(s) in E':
- $(j^q, k^q; x/y)$, if $x \in \Sigma_q$ and $\nexists$ an $a_r \neq -$ in y, $r \neq q$, $1 \leqslant r \leqslant n$.
- $(j^q, k^q; x/y)$ and $(j^q, k^r; x/y)$, if $x \in \Sigma_q$ and $\forall r,\ r \neq q$, $\exists$ an $a_r \neq -$ in y, $1 \leqslant r \leqslant n$.

Step 3. For each vertex $v \in V'$ where only dashed edges are arriving and leaving, remove from E' dashed edges arriving and leaving v and then remove v from V'.

Step 4. For each vertex $v_i \in V'$, if there are multiple edges with the same label (say, x/y) leaving v_i then create a null vertex v_i^x in V', create in E' a solid bold edge with label $-/-$ from v_i to v_i^x, and move these multiple edges so that they are leaving v_i^x instead of v_i. Make all remaining solid edges leaving v_i bold. The resulting digraph is $G' = (V', E')$.

In the second phase of the proposed method, first the set T_0 of triples $(t_{ij}, t_{jp}, \langle -O_k, +O_h\rangle)$ is formed where (t_{ij}, t_{jp}) is a transition pair with a potential undetectable 1-shift output fault. Then, the digraph $G' = (V', E')$ is modified using the information in T_0 to form the digraph $G'' = (V'', E'')$ such that in any traversal of G'', any two adjacent edges of G that are covered by the traversal of G'' will not create a potential controllability or observability problem. T_0 is constructed as follows:

for each vertex $v_j \in V$ $(1 \leqslant v_j \leqslant n)$ **do**
  for each edge e_{ij} (say t_{ij}) $= (v_i, v_j; x_j/y_j)$
    entering vertex v_j **do**
    for each edge e_{jp} (say t_{jp}) $= (v_j, v_p; x_{j+1}/y_{j+1})$
      leaving vertex v_j **do**
    if for some output $a_k \in \Gamma_k$, a_k is in y_j XOR
      a_k is in y_{j+1} AND $x_{j+1} \notin \Sigma_k$, **then**
    add $(t_{ij}, t_{jp}, \langle -O_k, +O_h \rangle)$ to T_0,
    where h is the tester sending the input x_{j+1} to
      the SUT in t_{jp} and k is the tester involved
      in the shift.

When forming T_0, for each tester involved in a possible shift in the transition pair (t_{ij}, t_{jp}), a new triple will be added to T_0. The digraph $G'' = (V'', E'')$ is constructed using digraph $G' = (V', E')$ and T_0 as follows:

Step 1. For each $(t_{ij}, t_{jp}, \langle -O_k, +O_h \rangle)$ triple in T_0, identify the vertex v_j in V' and add a new vertex v_j^* in V'' (if one does not exist already).

Step 2. A dashed edge from v_j^* to v_j is inserted in E'', with the label "$t_{ij}, t_{jp}, \langle -O_k, +O_h \rangle$" for each $(t_{ij}, t_{jp}, \langle -O_k, +O_h \rangle)$ in T_0. Any additional triples $(t_{ij}, t_{jp}, \langle -O_r, +O_s \rangle)$ in T_0 with the same transitions t_{ij} and t_{jp} will have $\langle -O_r, +O_s \rangle$ appended onto the label of this edge. If there is a triple $(t_{lm}, t_{mq}, \langle -O_t, +O_u \rangle)$ in T_0 where, in order to traverse the transition pair (t_{lm}, t_{mq}), the dashed edge from v_j^* to v_j must be traversed as is traversed for (t_{ij}, t_{jp}), then an alternate label "$t_{lm}, t_{mq}, \langle -O_t, +O_u \rangle$" is added to this dashed edge.

Step 3. For each solid edge t_{ij} in E' whose label is not $-/-$, if t_{ij} leaves a certain vertex v in V' then it will leave the same vertex in V''. If t_{ij} is contained in some transition pair (t_{ij}, t_{jk}) in a triple in T_0, then edge t_{ij} will go to v_j^* in V'', otherwise it will go to v_j in V''. The resulting digraph is $G'' = (V'', E'')$.

Step 4. Find a rural Chinese postman tour (RCPT) over the set of bold edges of G'' that starts at the vertex representing the initial state of M and minimizes the number of both transitions and external coordination message exchanges. This is achieved by assigning a sufficiently high cost to each dashed edge so that an RCPT over the bold edges avoids dashed edges whenever possible.

By construction, two transitions of G that would follow each other in any traversal of G'' will create neither a controllability nor an observability problem. Indeed, an RCPT over the set of bold edges in G'' will yield a minimum-length synchronizable global test sequence with no potential undetectable 1-shift output faults. This claim is justified as follows: after the first phase of the proposed method, any two solid edges (whether they have an edge with label $-/-$ between them or not) in the digraph G' that represent two adjacent transitions of the digraph G form a synchronizable sequence of transitions, either through their own properties (i.e., they are adjacent in G') or through the insertion of an external coordination message exchange relating to controllability between them. A global test sequence generated from G' would therefore be synchronizable. By a similar argument, after the second phase of the proposed method, any two solid edges (whether they have an edge with label $-/-$ between them or not) in the digraph G'' that represent two adjacent transitions of the digraph G form a synchronizable sequence of transitions without any potential undetectable 1-shift output faults, either through their own properties (i.e., they do not create an observability problem) or through the insertion of an external coordination message exchange relating to observability between them. A global test sequence generated from G'' would therefore be synchronizable, and would have no potential undetectable 1-shift output fault.

Fig. 2 shows the digraph G' for the $2p$-FSM M represented by the digraph G in Fig. 1. In all the $2p$-FSMs given in this paper, the two testers are called U (for upper) and L (for lower). Fig. 3 shows the

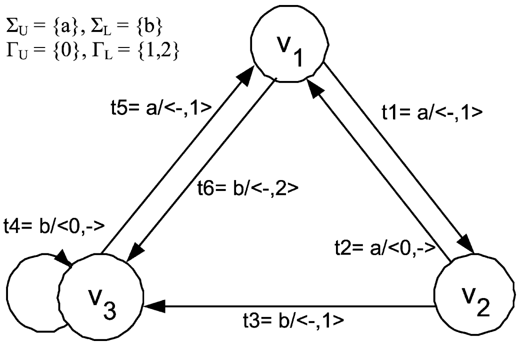

Fig. 1. Digraph G of $2p$-FSM $M1$.

digraph G'' constructed from the digraph G' using $T_0 = \{(t1, t2, \langle -O_L, +O_U \rangle), (t2, t1, \langle -O_L, +O_U \rangle), (t2, t6, \langle -O_U, +O_L \rangle), (t3, t4, \langle -O_U, +O_L \rangle), (t4, t5, \langle -O_L, +O_U \rangle), (t6, t4, \langle -O_U, +O_L \rangle)\}$. An RCPT over the bold edges in G'' in Fig. 3 yields a minimum-length synchronizable global test sequence that has no potential undetectable 1-shift output faults as the label of: $t1, t3, \langle -O_U, +O_L \rangle, t4, \langle -O_L, +O_U \rangle, t5, t6, \langle -O_U, +O_L \rangle, t4, \langle -O_L, +O_U \rangle, t5, t1, \langle -O_L, +O_U \rangle, t2$, for a total of five external coordination message exchanges relating to observability and nine input/output pairs.

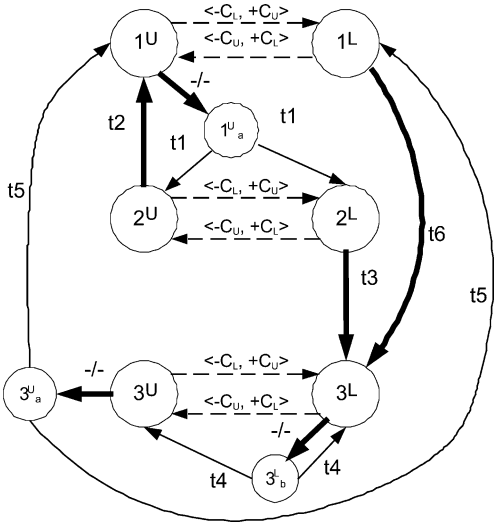

Fig. 2. Digraph G' of $2p$-FSM $M1$.

Although computing an RCPT over a subset of edges of a given directed graph is known to be NP-complete [8], a sufficient condition for the existence of a polynomial time algorithm is given in [1]. Accordingly, if the subgraph induced by the set of bold edges in G'' is weakly connected, then finding an RCPT over this set takes polynomial time. Otherwise, heuristics for RCPT are used [5].

It is important to note that post-processing of the RCPT over a subset of edges in G'' obtained in Step 4 provides two potential gains: The first one is the elimination of any external coordination message exchange from the resulting global test sequence that relates to a potential 1-shift output fault that can be rendered detectable by some existing subsequence in the global test sequence. The second one is the elimination of the need for explicit coordination among testers when the given FSM has an intrinsically synchronizable global test sequence. In this case, the post-processing examines the RCPT for occurrences of external coordination message exchanges relating to observability and eliminates each such message exchange inserted between a pair of transitions by appending a test subse-

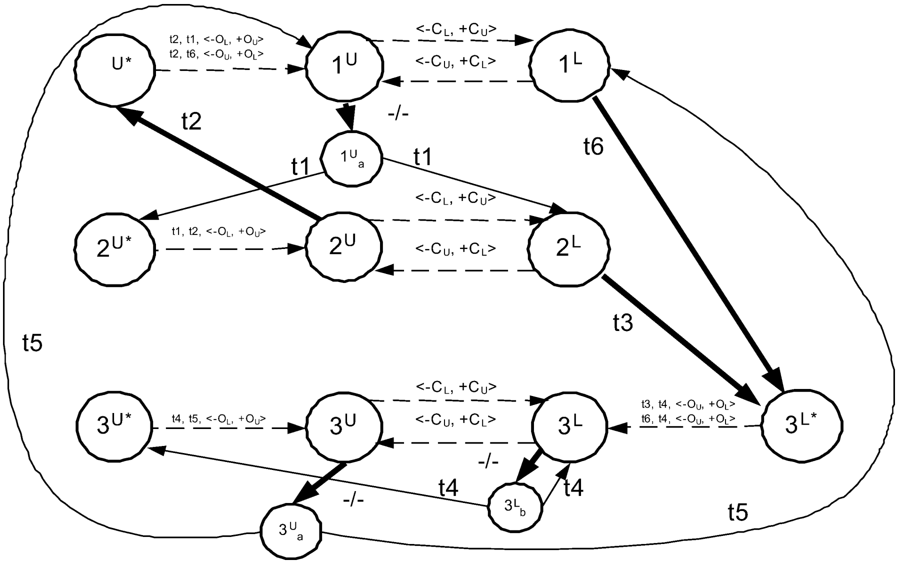

Fig. 3. Digraph G'' of $2p$-FSM $M1$.

quence to the RCPT to ensure that the corresponding potential 1-shift output fault is rendered detectable.

3.2. Comparison with other methods

The proposed method considers the same set $\Phi(M)$ of implementations of a given np-FSM M as the existing methods represented here by [3,14] for the construction of a global test sequence that can be applied without encountering any controllability and observability problems in a distributed test architecture. It guarantees the insertion of the minimum number of external coordination messages that need to be used to eliminate the potential controllability and observability problems, and thus compares favorably to the method of [3]. Indeed, when it is possible to construct a global test sequence for an FSM without inserting any external coordination message exchanges, the proposed method succeeds in constructing such a sequence whereas the method of [3] does not. For example, let us consider the digraph G of a $2p$-FSM $M2$ shown in Fig. 4. Any of 24 possible Chinese postman tours of this FSM may be the input to the method of [3]. Note that although the digraph of Fig. 4 contains no controllability problems, a global test sequence based on any of these Chinese postman tours will result in observability problems. This is because any instance of $t1$ or $t2$ followed directly by $t4$ or $t5$ (or vice versa) will result in a potential undetectable 1-shift output fault. The method of [3] starts with one Chinese postman tour out of 24 alternatives and requires the insertion of an external coordination message exchanges relating to observability into the selected tour to construct a synchronizable global test sequence with no potential 1-shift output fault. In contrast, the proposed method constructs a global test sequence where there is no potential controllability or observability problems as the label of $t1, t3, t4, t6, t2, t3, t5$ without using any external coordination message exchanges relating to controllability or observability.

Recall that the method in [14] starts with a given intrinsically synchronizable test sequence and examines this sequence for the instances of potential observability problems in an attempt to eliminate them in the given sequence by appending a test subsequence for each potential observability problem. Thus, this method is only applicable to FSMs which have one or more intrinsically synchronizable test sequences. In contrast, the proposed method is applicable to all FSMs as discussed above. In the case of an np-FSM for which one or more intrinsically synchronizable test sequences exist, the proposed method guarantees constructing one with no potential observability problem and with minimal length whereas the method in [14] does not guarantee the construction of one with minimal length. For example, let us consider the digraph G of a $2p$-FSM $M3$ shown in Fig. 5. The proposed method constructs an RCPT over the bold edges of G' which yields a minimum-length synchronizable global test sequence with no potential observability problem as the label of: $t3, t5, t6, t11, t10, t9, t7, t1, t2, t6, t9, t8, t5, t6, t11, t12, t4$ for a total of 17 input/output pairs. For the same FSM, the method of [LB94] constructs a synchronizable global test sequence with no potential observability problem as the label of: $t3, t5, t6, t11, t10, t9, t7,$

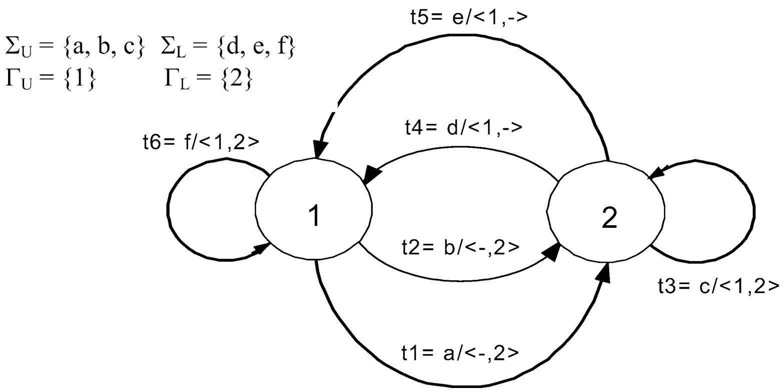

Fig. 4. Digraph G of $2p$-FSM $M2$.

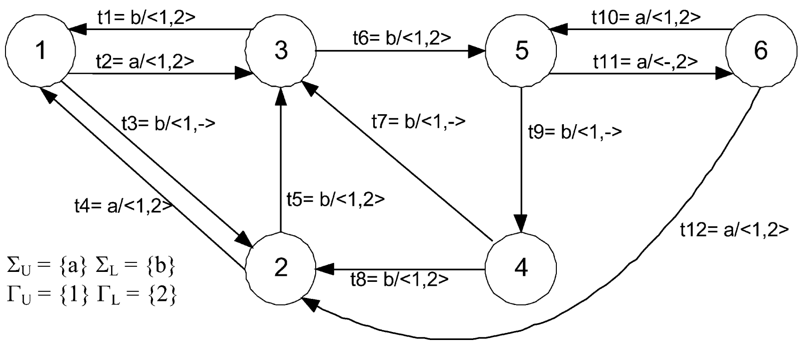

Fig. 5. Digraph G of $2p$-FSM $M3$.

$t1, t2, t6, t11, t12, t5, t6, t9, t8, t5, t6, t11, t12, t4$, for a total of 21 input/output pairs.

4. Conclusions

For distributed testing, a method has been proposed which constructs a minimum-length global test sequence that can be applied in a distributed test architecture without encountering any controllability and observability problems. If the given np-FSM has an intrinsically synchronizable global test sequence with no potential undetectable 1-shift output fault, the proposed method will construct such a sequence of minimum-length without inserting any external coordination message exchange relating to controllability and observability. If external coordination message exchanges are needed, then the proposed method generates a synchronizable global test sequence without any potential controllability and observability problems such that the number of both input/output pairs and external coordination message exchanges are minimized.

In this paper, in addition to the controllability problem, the observability problem that stems from 1-shift output faults is addressed. There are two related problems that remain open. These are the problem of finding a minimum-length test subsequence to replace an external coordination message exchange relating to observability and the problem of constructing a synchronizable global test sequence with no potential k-shift output faults where $k \geqslant 2$. It is interesting to note that the solution for a special case of the second problem is also provided by the method proposed in this paper. It is shown in [19] that given a synchronizable global test sequence for a $2p$-FSM where all 1-shift output faults are detectable when applied in the distributed test architecture (either through insertion of an external coordination message exchange related to observability or not), and where each output associated with a particular transition within the global test sequence is unique, there will be no undetectable k-shift output faults, $k \geqslant 2$, in the global test sequence. However, if the assumption of output uniqueness is removed, then it can easily be shown that certain k-shift output faults may remain undetectable. Hence, the general case remains open.

References

[1] A. Aho, A. Dahbura, D. Lee, M. Uyar, An optimization technique for protocol conformance test generation based on UIO sequences and rural Chinese postman tours, IEEE Trans. Comm. 39 (11) (1991) 1604–1615.

[2] S. Boyd, H. Ural, The synchronization problem in protocol testing and its complexity, Inform. Process. Lett. 40 (1991) 131–136.

[3] L. Cacciari, O. Rafiq, Controllability and observability in distributed testing, Inform. Software Technol. 41 (1999) 767–780.

[4] W. Chen, H. Ural, Minimum-cost synchronizable test sequences based on multiple UIO sequences, IEEE/ACM Trans. Networking 3 (1995) 152–157.

[5] J. Edmonds, E.L. Johnson, Matching, Euler tours and the Chinese postman, Math. Programm. 5 (1973) 88–124.

[6] A. En-Nouaary, R. Dssouli, F. Khendek, Timed Wp-method: Testing real-time systems, IEEE Trans. Software Engrg. 28 (11) (2002) 1023–1038.

[7] K. Farooqui, L. Logrippo, J. de Meer, The ISO reference model for open distributed processing: an introduction, Comput. Networks ISDN Syst. 27 (1995) 1215–1229.

[8] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of Intractability, Freeman, New York, 1979.
[9] S. Guyot, H. Ural, Synchronizable checking sequences based on UIO sequences, in: IFIP IWPTS'95, Evry, France, 1995, pp. 395–407.
[10] R.M. Hierons, Testing a distributed system: generating minimal synchronised test sequences that detect output-shifting faults, Inform. Software Technol. 43 (9) (2001) 551–560.
[11] ISO/IEC Information technology—Opens Systems Interconnection, 9646, Parts 1–7, 1995.
[12] ISO/IEC Open Distributed Processing, Reference Model, 10746, Parts 1–4, 1996.
[13] D. Lee, M. Yannakakis, Principles and methods of testing finite state machines: A survey, Proc. IEEE 84 (8) (1996) 1090–1123.
[14] G. Luo, R. Dssouli, G.v. Bochmann, P. Venkataram, A. Ghedamsi, Test generation with respect to distributed interfaces, Comput. Standards Interfaces 16 (1994) 119–132.
[15] B. Sarikaya, G.v. Bochmann, Synchronization and specification issues in protocol testing, IEEE Trans. Comm. 32 (1984) 389–395.
[16] K.C. Tai, Y.C. Young, Synchronizable test sequences of finite state machines, Comput. Networks ISDN Syst. 13 (1998) 1111–1134.
[17] H. Ural, Z. Wang, Synchronizable test sequence generation using UIO sequences, Comput. Comm. 16 (1993) 653–661.
[18] C. Wang, M. Schwartz, Fault detection with multiple observers, IEEE/ACM Trans. Networking 1 (1993) 48–55.
[19] D. Whittier, Solutions to controllability and observability problems in distributed testing, Thesis, SITE, University of Ottawa, ON, 2001.
[20] W.-J. Wu, C.Y. Tang, Synchronizable test sequence for multi-party protocol conformance testing, Comput. Comm. 21 (1998) 1177–1183.
[21] Y.C. Young, K.C. Tai, Observation inaccuracy in conformance testing with multiple testers, IEEE WASET (1998) 80–85.