Characterizing Faults on Real-Time Systems Based on Grid Automata

Adilson Luiz Bonifacio* and Gilson Doi

Computing Department, University of Londrina P.O. Box 6.001, 86.051-980, Londrina, PR, Brazil

Abstract: Real-time systems are, in general, critical systems that interact with the environment through input and output events regulated by time constraints. The testing activity on systems of this nature requires rigorous approaches due to their critical aspects. Model-based testing approaches rely on formalisms that provide more reliability to testing activities. However, a model-based testing approach for real-time systems depends on techniques that can deal with continuous evolution of time appropriately. Several testing approaches apply discretization techniques in order to represent continuous behavior of timed models. Test suites can then be extracted from discretized models to support conformance testing between specifications and their respective implementations. Therefore an evaluation of test suites considering a fault coverage is an important task, but rarely addressed by model-based testing approaches for real-time systems. In this work we propose a systematic strategy to identify faults in TIOA models based on their corresponding discretized models. We precisely define a fault model to support model-based testing activities such as coverage analysis and test case generation.

Keywords: Real-time systems, Model-based testing, TIOA, Grid automata, Fault model.

1. INTRODUCTION

Critical systems [16], where a failure can cause severe or irreparable damage, require rigorous methods to support activities of software development process, especially the activity of testing. Model-based testing [3, 19] is an approach that relies on well defined formalisms to specify and to test system requirements. These formalisms avoid misinterpretations on system requirements and provide more reliability to the product by reducing drastically failures in the testing procedures [14].

Although model-based testing is a promising approach and largely studied its application to deal with continuous time evolution when modeling real-time systems [15, 17] is still a challenge. A typical problem that can arise on validating systems of this nature is the combinatorial explosion on their state space representation. The system state space grows exponentially as the number of state variables that model the continuous behavior increases in the system [7].

Some model-based testing approaches for real-time systems [4, 9, 10, 18] apply discretization techniques to cope with the continuous time evolution. Discretized models allow us to specify system requirements with time aspects in a finite state space but still exponential. Testing activities such as test suite generation, conformance verification and fault coverage analysis can be performed based on these discretized models.

The conformance relation between an implementation and its specification can be verified according to a fault model [5, 17]. A fault model [6, 11] gives the basis to detect faults and also to evaluate the capability of a certain test suite on finding all faults due to disagreements between specifications and their respective implementations.

This work proposes a fault model based on grid automata which, in turn, are obtained by discretization methods for Timed Input/Output Automata (TIOA). The TIOA model specifies real-time systems that interacts with the environment. This fault model aids to identify classes of fault on TIOA models through their corresponding grid automata. Testing activities can then be performed using grid automata such as a fault coverage analysis of test suites for real-time systems.

This paper is organized as follows. In Section 2 some related works are summarized. The definition of fault model and the fault models for Finite State Machines (FSM) and TIOA models are presented in Section 3. Section 4 generalizes the representation of the continuous time evolution on grid automata. Such generalization then supports the fault characterization following the fault model described in Section 5. The concluding remarks and some future directions are given in Section 6.

2. RELATED WORK

Some model-based testing approaches have been applied on real-time systems using their discretized

*Address correspondence to this author at the Computing Department, University of Londrina P.O. Box 6.001, 86.051-980, Londrina, PR, Brazil; Tel:+55(43)3371-4678; Fax: +55(43)3371-4294;
E-mail: adilsonbonifacio@gmail.com

models to specify TIOA models. However, testing approaches based on TIOA have been lacking due to time aspects and their reactive nature. Test generation methods based on TIOA model generally derive large test suites that are unfeasible in practice. Some works have been proposed to obtain more compact test suites by establishing discretization techniques. Considering practical limitations on testing real-time systems, fault characterization and identification techniques have been used on discretized models.

Springintveld *et al.* [18] propose a discretization approach *where* the granularity is based on the number of clock variables present in a TIOA model. Test suites are obtained applying an extension of the W-method [6] which generates extremely large test suites in size. They use a tight relation between the chosen granularity and the number of clocks. Thus the state space grows quickly according to the number of clocks which becomes unfeasible in practice. Their approach does not accomplish a precise fault coverage analysis, particularly for timing faults.

A timed extension of the Wp-method [12], called Timed-Wp method [9] for real-time systems, generates discretized models based on the number of clock variables. Thereby it results again on a exponentially large state space since clock regions are obtained according to the number of clocks comprised in the TIOA model. Furthermore the fault coverage analysis is conducted according to a fault model based on region automata [11]. Again, it causes the state space problem which hinders a precise fault identification process and the coverage analysis.

Other works [8, 10] deal with discretization approaches where granularities are chosen on fixed points obtained by traversing guards and invariant conditions over the TIOA. In these approaches the state space of discretized models becomes more manageable and they lean on the number of transitions in the original model. However few fixed points over the enabled time interval of the transition may not be reasonable to represent the continuous time evolution in a general case. Further there is no mechanism to systematically detect faults in these approaches.

In a more recent work [4] a testing framework is proposed to discretize TIOA models. This approach allows for more flexible choices of granularities but still guarantees that the original model simulates homomorphically the corresponding grid automaton and conversely. In this case the state space becomes more manageable which opens the possibility for applying this method in practical experiments. However the fault coverage analysis is based on test purposes which restrains the fault detection and hence the efficiency of this method.

Devising more efficient methods of testing that deal with real-time systems we propose a fault model to TIOA formalism based on grid automata. The aim is to characterize classes of potential faults in implementations modeled by TIOA models. The proposed fault model supports the fault detection for real-time systems based on grid automata.

3. FAULT MODELS

The fault detection analysis results from the application of test suites to computational systems. The efficiency of a fault detection or the accuracy of a test suite to finding faults depends on systematic strategies to allow test automation. Then the concept of fault must be well-defined in a specific scope to automating model-based testing approaches.

The fault detection process can be classified in two analyses: quantitative or qualitative. A fault detection based on fault models [6, 11] addresses an quantitative analysis whereas test purposes are considered qualitative. A fault model classifies potential faults in a system under test where each class of faults can be generally characterized to detect any fault within a particular scope. In contrast, test purposes detect particular faults and fall short for finding general faults. So an approach based on a fault model results more effectiveness when finding faults on systems. Note that specific faults of a system are encompassed by faults detected using a more general fault model under the same test assumptions.

The coverage analysis of the W-method [6] follows a fault model defined for FSMs. A FSM is a formalism widely applied due to its plainness and expressiveness on modeling systems [13, 20]. That work classifies the fault model in classes of faults: operation faults, transfer faults and faults of extra/missing states. An operation fault occurs when the operation function gives rise an output action in the specification distinct to that yielded by the implementation. A transfer fault occurs when a transfer function leads the implementation to a distinct state from that modeled in the specification. Faults of extra/missing states occur when the implementation may be changed by adding or removing states to be in conformance to the specification. Nonetheless FSMs do not specify continuous time evolution and, further, input and output actions are strongly associated which means reactive aspects can not be captured appropriately by the formalism.

A TIOA can model continuous time evolution and also reactive aspects where output actions produced by systems are uncontrolled. A TIOA [1, 2] is formally defined by a tuple $M = (S, s_0, \Sigma, C, v_0, Inv, T)$, where S is a set of states, s_0 is the initial state, and $\Sigma = X \cup Y$ is a set of actions. In order to capture the reactive behavior, Σ is partitioned in a set of input actions X and in a set of output actions Y with $X \cap Y = \varnothing$. A set of clock variables C describes the continuous time evolution through the notion of clock interpretations. The initial clock interpretation is given by v_0. The set of invariant conditions $Inv : S \to \Phi_C$ is defined as a mapping from the set of states S to the set of clock constraints Φ_C. A TIOA keeps in a state while its invariant condition is satisfied continuously. The set of transitions is defined by $T \subseteq (S \times \Sigma \times \Phi_C \times [C \curvearrowright \mathbb{Q}_{\geq}] \times S)$ where $[C \curvearrowright \mathbb{Q}_{\geq}]$ is a reset operation that maps a set of clocks to their respective values in the domain of non-negative rationals. Note that clock variables evolve synchronously unless a reset operation occurs over the clock variables. So the meaning for a transition $(s, z, \delta, \theta, r)$ is that the machine can move to state r from state s over the symbol z provided that the guard δ is enabled. Further, upon moving to state r, the mapping $\theta \in [C \curvearrowright \mathbb{Q}_{\geq}]$ indicates which clocks are reset and to which values. The set of all clock conditions, Φ_C, is comprised by all expressions δ that can be finitely generated using the rules $\delta := \text{true} \mid c \leq \tau \mid \tau \leq c \mid \neg\delta \mid \delta_1 \wedge \delta_2$, where c is a clock variable and $\tau \in Q_{\geq}$ is a time instant. A clock interpretation over C is a partial function from C into $Q_{\geq}$. A total clock interpretation over C is a clock interpretation over C whose domain is C. The set of all clock interpretations over C is denoted by $[C \curvearrowright \mathbb{Q}_{\geq}]$, and $[C \to Q_{\geq}]$ denotes the set of all total clock interpretations over C. A precise definition of the TIOA semantics can be found in [4].

En-Nouaary et. al. [9] proposes a fault model for TIOA based on region automata to support the coverage analysis using the Timed Wp-method. A region automaton is extracted from a TIOA model based on the number of clock regions [1]. A state of a region automaton is composed by a respective state in the corresponding TIOA and a reachable clock region. Transitions are labeled with actions according to the reachability analysis over clock regions of the TIOA. A fault is detected when an implementation is modeled by a distinct region automaton of its specification. The proposed fault model can detect faults in a TIOA generating different region automatons based on the specification region automaton.

In their work faults are classified in non-timing faults and timing faults. Non-timing faults resemble operation and transfer faults proposed by [6] on FSM models. On the other hand, timing faults are closely related to the semantics of continuous time evolution and they can be classified in restriction faults of clock conditions, widening faults of clock conditions and reset faults.

A restriction fault of a clock condition occurs in a transition with an input action that implements a tighter clock condition. Otherwise, a tighter condition in a transition with output action does not incur in a restriction fault because output actions are uncontrolled and since the enabled time interval is included in the specified time interval. On the other hand, a widening fault occurs in a transition that implements a more relaxed clock condition. Note that restriction and widening faults over clock conditions change the reachable clock regions of the implementation. Thus those faults can be detected by test cases that cover all states of region automaton. Faults of reset clocks occur when the set of clocks to be reset in a specified transition is not correctly implemented. These faults are also classified in two cases, when a specified reset is not implemented and some clocks keep evolving on time or when a non specified reset is implemented in such a way that boundary values of the system are not explored. Their work also assumes a special action *ResetClock* to set up a clock reset. This action identifies clock resets in implementations and also which clocks are reset. Faults of clock resets can be then detected by the occurrence of the *ResetClock* action. However an implementation under test is a black-box and thus, in general, reset actions can not be observed in practical applications. The reset fault detection in their approach is highly limited to a specific scope where implementations enclose observable actions on clock resets.

Furthermore, the fault model proposed by En-Nouaary et. al. [9] does not provide the identification of every class of faults. Verdicts of faults over implementations are not enough to identify the fault class based on a specific fault. Then their approach only allows the fault detection by pointing out a disagreement between an implementation and its specification given by their respective region automatons.

4. GENERALIZING THE TIME DISCRETIZATION

The proposed fault model is based on grid automata obtained by discretizing TIOA specifications. A grid automaton is composed by a set of states and a set of discrete transitions. A transition in a grid can be labeled with an input or output action that represents a discrete transition in the corresponding TIOA or it can be labeled with a granularity to represent the time evolution. Then a grid simulates its corresponding TIOA through a sequence of movements that represents the interacting with the environment and also the time evolution [4].

A more generalized analysis of a TIOA requires a generalization over the grid representation. Such generalizations support the characterization of timing faults described in Section 5.

A discretization of the continuous time evolution in a state s of a TIOA derives a infinite number of configurations. Configurations are pairs of states and clock interpretations that give rise to corresponding grid states according to the chosen granularity. Then a set of grid states are generated from s with different clock interpretations. Thus a sequence of $n \in \mathbb{N}$ grid transitions labeled with $g \in \mathbb{Q}_{\geq}$ models the continuous time delay $n \cdot g$ in the state s. This sequence of grid transitions is called *time line*. A *maximal time line* is the sequence of transitions that models the greatest time delay starting at a reachable state s up to the upper bound of its invariant condition. The notion of *time line* allows us to identify whether a set of grid states represents a continuous time interval within a TIOA state.

The enable time interval in TIOA transitions is also an aspect that must be generalized in the discretization process. A transition t of a TIOA is enabled in a time interval while clock conditions are satisfied on the transition. The TIOA transition is modeled by a set of grid transitions T' labeled with the same action for all time interval where conditions are satisfied since there are no clock conditions in the grid. Then there is a transition in T' for each grid state corresponding to a configuration that enables the clock condition at t. As more relaxed is the enabling time interval in t more grid transitions are required in the grid to model t, whereas tighter time intervals in t give rise to a smaller number of transitions in the grid.

When an action of a transition occurs in a TIOA, clock interpretations can also be changed by reset functions. A set of resets in a transition may change the target states of this transition representation in the grid. The grid representation of clock resets can be generalized by three cases: when there is no clock to be reset; when a proper subset of clocks is reset; and if all clocks in the TIOA is reset. In the first case the clock interpretation at target state must be the same when the action occurs. Hence the TIOA transition is modeled by two transitions going out of the states obtained on the subsequent time instants. Similarly, it also takes place at target states of those transitions. In the second case for all time interval whenever the action can occur in the TIOA will result on a different combination of all clock interpretations. Hence all possible combinations over clock variables result in several *time lines* at the target state. In the last case the configuration at the target state of the transition is the same for all time interval whenever the action can occur according to the clock conditions in the TIOA. Then each transition in the grid that models the TIOA transition leads to the same single target state in the grid.

5. CHARACTERIZING FAULTS BASED ON GRID AUTOMATA

A more precise coverage analysis allows us to finding faults in a system using a fault model. A fault model drives the identification process over candidate implementations according to a particular specification.

In the fault model proposed in [11] region automata are used to characterize timing faults on TIOA models. The coverage analysis for the *Timed-Wp* method is based on region automata. However, it does not precisely identify classes of faults over region automata. As known a infinite number of equivalent clock interpretations is represented by a single clock region. Therein it is not possible get the precise time instant whenever a fault occurs.

A state of a grid automaton is derived from a TIOA configuration and composed by a TIOA state and a clock interpretation. Therefore a detection analysis based on grid automata can identify not only its fault class but also the precise time instant when the fault occurs. In this section we propose a fault model based on grids obtained by more flexible discretizations [4]. Thus, our detection approach can be applied to discretized models based on clock regions. In contrast, however, the fault model based on region automata can not deal with grid automata and so the fault detection can not be applied to grids derived from flexible discretizations.

This fault model characterizes the classes of fault for TIOA models using grid automata. We formally define a fault analysis for timed models. We then classify faults in two main groups: non-timing and timing faults. Non-timing faults are characterized by action and transfer faults whereas timing faults lead to restriction and widening faults of clock conditions as well as faults of clock reset.

Similarly to other approaches we assume some test hypothesis to set out a detection scope and to characterize these classes of fault.

Assumption 1 The fault model assumes that:

1. the alphabet of actions are the same in the implementation and the specification;
2. the implementation behaviors can be modeled by a TIOA g-adjusted and L-bounded [4];
3. the TIOA specification and implementation are complete, that is, for every state there is only one transition for each input action;
4. the TIOA specification is deterministic, that is, for every state there is a unique transition for each input action;
5. the TIOA specification is isolated output, that is, for every state only one transition is labeled with an output action.

5.1. Non-Timing Faults

Non-timing faults occur on syntactical elements of transitions and actions in a TIOA. An action fault occurs when a transition in the set T' which represents a TIOA transition, as shown in Section 4, is labeled by a distinct action with respect to the specification. We guarantee that the observable action in the implementation does not correspond to another transition from this state since the TIOA is output isolated. Also for every input action from a single state of the implementation there must be a transition with the same input action in the specification since the TIOA is complete. Hence there is no action fault in transitions labeled by input actions. Definition 1 formalizes the characterization of action fault.

Definition 1. Let $M = (S, s_0, \Sigma, C, v, Inv, T)$ be a specification TIOA and let $M_G = (S_G, s_G, \Sigma_G, T_G)$ be the corresponding grid. Also let $M'_G = (S'_G, s'_G, \Sigma'_G, T'_G)$ be the grid automaton of a candidate implementation. Assume that $T' \subseteq T_G$ is the set of transitions representing a transition $t = (s_i, \sigma, \delta, \theta, s_j) \in T$. Let $(q_i, \sigma, q_j) \in T'$ and let Ψ be a set of grid words such that for each $\psi \in \Psi$ we get $s_G \overset{\psi}{\Vdash} q_i$ and $s'_G \overset{\psi}{\Vdash} q'_i$, where $q'_i \in S'_G$. A action fault takes place in t when for at least one $\psi \in \Psi$, there exists $(q'_i, \boldsymbol{\sigma'}, q'_j) \in T'_G$, where $q'_j \in S'_G$, $\boldsymbol{\sigma'} \in \boldsymbol{\Sigma}'_G \setminus g$ and we have that $\sigma' \neq \sigma$.

A transfer fault occurs when a transition is implemented by a distinct target state with respect to the specification. This fault is characterized on grids by using the notion of characterization set, a set of input sequences that is able to identify states on the model [6, 12]. In a transition of a TIOA represented by the set T', as described in Section 4, the transfer fault is identified by applying a characterization set in the target state for each transition in T'. The transfer fault is then characterized when the sequence of observable outputs in the implementation is distinguished from the specification. Definition 2 formalizes the transfer fault.

Definition 2. Let $M = (S, s_0, \Sigma, C, v, Inv, T)$ be a specification TIOA and let $M_G = (S_G, s_G, \Sigma_G, T_G)$ be the corresponding grid. Also let $M'_G = (S'_G, s'_G, \Sigma'_G, T'_G)$ be the grid automaton of a candidate implementation. Assume that $T' \subseteq T_G$ is the set of transitions representing a transition $t = (s_i, \sigma, \delta, \theta, s_j) \in T$. Let $(q_i, \sigma, q_j) \in T'$ and let Ψ be a set of grid words such that for each $\psi \in \Psi$ we get $s_G \overset{\psi}{\Vdash} q_j$ and $s'_G \overset{\psi}{\Vdash} q'_j$ with $q'_j \in S'_G$, and W is a characterization set to M_G. A transfer fault takes place in t when for every $\psi \in \Psi$, there is a $w_k \in W$ such that $s'_G \overset{\psi\cdot\omega'}{\vDash} p'_i$ and $s_G \overset{\psi\cdot\omega}{\vDash} p_i$, where $p_i \in S_G$, $p'_i \in S'_G$, ω and ω' are grid words with $w_k = (\omega \downarrow_Y) = (\omega' \downarrow_Y)$ and $(\omega \downarrow_X) \neq (\omega' \downarrow_X)$.

5.2. Timing Faults

Timing faults occur when disagreements arise from the semantics of continuous time evolution in a TIOA. These faults can be detected on clock conditions and clock resets. A fault of clock condition is also classified into restriction and widening faults.

A restriction fault of clock condition occurs when a specified TIOA condition is tighter implemented. Then this implemented transition may not be enabled for some time instants that are allowed in the specification. The detection of restriction fault on clock conditions in a transition t occurs by applying a characterization set on the target state of each transition within the set T' which represents t. The detection of this fault resembles the detection of transfer faults since an input action is allowed in any state of the model. Thus a transition is taken with a distinction input action on the grid implementation whenever such transition is not enabled in a time interval. Definition 3 formalizes the restriction fault of clock conditions.

Definition 3. Let $M = (S, s_0, \Sigma, C, \nu, Inv, T)$ be a specification TIOA and let $M_G = (S_G, s_G, \Sigma_G, T_G)$ be the corresponding grid. Also let $M'_G = (S'_G, s'_G, \Sigma'_G, T'_G)$ be the grid automaton of a candidate implementation. Assume that $T' \subseteq T_G$ is the set of transitions representing a transition $t = (s_i, \sigma, \delta, \theta, s_j) \in T$. Let $(q_i, \sigma, q_j) \in T'$ and let Ψ be a set of grid words such that for each $\psi \in \Psi$ we get $s_G \overset{\psi}{\Vdash} q_j$ and $s'_G \overset{\psi}{\Vdash} q'_j$ with $q'_j \in S'_G$, and W is a characterization set to M_G. A restriction fault of clock conditions takes place in t when there exists a $\Psi' \subseteq \Psi$ such that for every $\psi' \in \Psi'$, there is a $w_k \in W$ such that $s'_G \overset{\psi' \cdot \omega'}{\models} p'_i$, $s_G \overset{\psi' \cdot \omega}{\models} p_i$, where ω' and ω are grid words with $w_k = (\omega' \downarrow_Y) = (\omega \downarrow_Y)$ and $(\omega \downarrow_X) \neq (\omega' \downarrow_X)$.

We note that restriction faults over clock conditions are not considered for output actions since they are not under control of the tester. Output actions are autonomously yielded by the system under test. Thereby when these actions occur in any time instant within the time interval of the specification they will also be deemed correct in the corresponding implemented transition.

The widening fault of clock conditions, otherwise, occurs when a specified clock condition is more relaxed implemented. In this case the implemented transition is enabled in a time interval that was not formerly allowed in the specification. A widening fault of clock conditions in a transition t is given by extra transitions labeled with output action on the grid implementation. Let this extra transition be labeled by σ and going out of a state p to any other state in the grid implementation. A widening fault can be detected through the TIOA state that is represented by p because the grid is output isolated. Then a widening fault over clock conditions at the TIOA transition going out of this state with σ is then characterized by an extra transition. We note that if there is no transition in the TIOA specification going out of the state with σ, *i.e.* a *false* condition, we have a relaxed condition at a transition that is not enabled in the specification. Definition 4 formalizes the widening fault of clock condition.

Definition 4. Let $M = (S, s_0, \Sigma, C, \nu, Inv, T)$ be a specification TIOA and let $M_G = (S_G, s_G, \Sigma_G, T_G)$ be the corresponding grid. Also let $M'_G = (S'_G, s'_G, \Sigma'_G, T'_G)$ be the grid automaton of a candidate implementation. Assume that $T' \subseteq T_G$ is the set of transitions representing a transition $t = (s_i, \sigma, \delta, \theta, s_j) \in T$. Let $(q_i, \sigma, q_j) \in T'$ and let Ψ be a set of grid words such that for each $\psi \in \Psi$ we get $s_G \overset{\psi}{\Vdash} q_i$ and $s'_G \overset{\psi}{\Vdash} q'_i$ with $q'_i \in S'_G$, Ψ' a set of grid words such that for each $\psi' \in \Psi'$ we get $s_G \overset{\psi'}{\Vdash} q$ and $s'_G \overset{\psi'}{\Vdash} q'$ with $q \in S_i$ e $q' \in S_G$. A widening fault of clock conditions takes place in t when there exists at least one $\psi'' \in \Psi''$ such that $s'_G \overset{\psi'' \cdot \sigma}{\models} p'_i$ and $s_G \overset{\psi'' \cdot \sigma}{\models} p_i$, with $\sigma' \in Y \cup \{g\}$ and $\Psi'' = \Psi' - \Psi$, and we have $\sigma \neq \sigma'$.

The fault of clock reset occurs by a faulty implementation on the reset functions. Inappropriate implementations of reset functions give rise to unexpected behaviors that can be detected by a distinct reachable configuration at target state on TIOA specification.

A reset fault occurs at a transition t of the TIOA if for every transition in T' there is a distinct target state in the grid specification. The target state related to each clock interpretation is checked through the obtained behavior since the implementation is a black-box. However, a characterization set, in this case, can not aid the detection process. Although target states in the grid implementation are distinguished from the grid specification they will represent the same single state in the TIOA. So there is no guarantee that the characterization set distinguishes those states.

In order to overcome this problem we apply the detection approach of reset faults on grids based on the notion of fault propagation. Although a reset fault in a transition can not be straight detected on grids they can be identified by other classes of fault. Assume a TIOA transition $t_i = (s, \sigma_i, \delta_i, \theta_i, s')$ with a reset fault over a clock variable c_i and a set T_t of transitions in the TIOA that can be enabled after t_i. Let $t_j = (r, \sigma_j, \delta_j, \theta_j, r')$ be a transition in T_t, a reset fault occurs in t_i if the reset $v(c_i) \oplus \theta_i(c_i)$ results in a clock interpretation distinct to that one specified. This fault is propagated for every t_j that is enabled after a sequence of movements when t_j does not reset c_i and δ_j is in the form $\tau_i \le c_i \le \tau_j$. Note that $c = \tau_i$ is represented by $\tau_i \le c_i \le \tau_j$ where $\tau_j = \tau_i$ and $\tau_i < c_i < \tau_j$ is discretized to $\tau_i + g \le c_i \le \tau_j - g$.

When a reset implementation $v(c_i) \oplus \theta_i(c_i)$ derives a clock interpretation greater than that one specified, the time interval where t_j is enabled in the implementation is smaller than that one established in the specification. Thus the clock condition at t_j is tighter on the upper bound. If the state r is reachable with a clock interpretation smaller than the lower bound of the clock condition that enables the transition t_j after t_i is taken, then the system continuously evolves in time at state r up to t_j occurs. When the reset $v(c_i) \oplus \theta_i(c_i)$ derives a clock interpretation greater than that one specified, the time interval will be smaller. Thus the clock condition at t_j is more relaxed on the lower bound. When a reset implementation $v(c_i) \oplus \theta_i(c_i)$ derives a clock interpretation smaller than that one specified, the time interval whenever t_j is enabled in the implementation is greater than in the specification. Thus the clock condition at t_j is more relaxed on the upper bound. Again, if the state r is reachable with a clock interpretation smaller than the lower bound of the clock condition that enables the transition t_j after t_i is taken, then the system continuously evolves in time at state r up to t_j occurs. When the reset $v(c_i) \oplus \theta_i(c_i)$ derives a clock interpretation smaller than that one specified, the time interval will be greater. Thus the clock condition at t_j is tighter on the lower bound. The propagation of reset faults is formalized in Definition 5.

Definition 5 Let $M = (S, s_0, \Sigma, C, v, Inv, T)$ and $M' = (S', s_0', \Sigma', C', v', Inv', T')$ be two TIOA, the specification and the candidate implementation, respectively. Assume $v(c_i) \oplus \theta_i(c_i) = k$, where $k \in \mathbb{Q}_{\ge}$ is a clock interpretation $v(c_i)$ after $t_i = (s_i, \sigma_i, \delta_i, \theta_i, r_i) \in T$ is taken, with δ_i in the form $\tau_i \le c_i \le \tau_j$ and that $T_t \subset T$ is the subset of transitions that are enabled after t_i, where each $t_j = (s_j, \sigma_j, \delta_j, \theta_j, r_j) \in T_t$ with $\delta_j = \tau_x \le c_i \le \tau_y$. Let ψ be a timed word such that $s_0 \overset{\psi}{\vDash} s_i$ and let Ω be a set of timed words where for every $\omega \in \Omega$, $s_i \overset{\omega}{\vDash} s_j$. Also assume that given the projection $\omega' = \omega \downarrow_{(X \cup Y)}$, with $\omega' = (\omega_1', \ldots \omega_n')$ we get $\Delta_k \in \mathbb{Q}_{\ge}$ with $\Delta_k = \omega_1' + \ldots + \omega_n'$. A reset fault occurs in t_i when $t_i' = (s_i', \sigma_i', \delta_i', \theta_i', r_i') \in T'$, such that $v(c_i) \oplus \theta_i'(c_i) = k'$, with $k' \in \mathbb{Q}_{\ge}$ and we have $k' \ne k$. Such fault is propagated for all $t_j' = (s_j', \sigma_j', \delta_j', \theta_j', r_j') \in T'$ with δ_j' in the form $\tau_i' \le c_i' \le \tau_j'$ if:

1. $k' < k$ and $k + \Delta_k < \tau_i$, then with $s_0 \overset{\psi \cdot \omega}{\vDash} s_j$ we get $v(c_i') = k' + \Delta_k$ with $k' + \Delta_k < k + \Delta_k$, $\tau_i - (k' + \Delta_k) > \tau_i - (k + \Delta_k)$ and $\tau_j - (k' + \Delta_k) > \tau_j - (k + \Delta_k)$, resulting in a tighter τ_i' and a more relaxed τ_j'.

2. $k' < k$ and $k' + \Delta_k > \tau_i$, then with $s_0 \overset{\psi \cdot \omega}{\vDash} s_j$ we get $v(c_i') = k' + \Delta_k$ with $k' + \Delta_k < k + \Delta_k$ and $\tau_j - (k' + \Delta_k) > \tau_j - (k + \Delta_k)$, resulting in a more relaxed τ_j'.

3. $k' > k$ and $k' + \Delta_k < \tau_i$, then with $s_0 \overset{\psi \cdot \omega}{\vDash} s_j$ we get $v(c_i') = k' + \Delta_k$, with $k' + \Delta_k > k + \Delta_k$, $\tau_i - (k' + \Delta_k) < \tau_i - (k + \Delta_k)$ and $\tau_j - (k' + \Delta_k) < \tau_j - (k + \Delta_k)$, resulting in a more relaxed τ_i' and a tighter τ_j'.

4. $k' > k$ and $k + \Delta_k > \tau_i$, then with $s_0 \overset{\psi \cdot \omega}{\vDash} s_j$ we get $v(c_i') = k' + \Delta_k$ with $k' + \Delta_k > k + \Delta_k$ and $\tau_j - (k' + \Delta_k) < \tau_j - (k + \Delta_k)$, resulting in a tighter τ_j'.

We notice that if there is no condition over the clock c_i in a transition $t_j \in T_t$ then fault propagation does not occur in t_j. When the clock condition is composed in a transition $t_j \in T_t$, in the form $\delta_i \vee \delta_j$, a reset fault is propagated to t_j if the representation of δ_i and δ_j shows fault propagation in a time interval with empty intersection between both conditions. Otherwise, when the clock condition is composed in the form $\delta_i \wedge \delta_j$, a reset fault is propagated to t_j if the representation of δ_i and δ_j shows fault propagation in a common time interval between both conditions. A fault propagation on δ_i and δ_j is given as in Definitions 5 and 6. Note that δ_i and δ_j can also be composed conditions.

Definition 6 Let $M = (S, s_0, \Sigma, C, v, Inv, T)$ and $M' = (S', s_0', \Sigma', C', v', Inv', T')$ be two TIOA, the specification and the candidate implementation, respectively. Assume that $T_t \subset T$ is the subset of transitions that are enabled after t_i. A reset fault occurs in t_i when $t_i' = (s_i', \sigma_i', \delta_i', \theta_i', r_i') \in T'$, such that $v(c_i) \oplus \theta_i'(c_i) = k'$, with $k' \in Q_{\geq}$ and we have $k' \neq k$. Such fault is propagated for all t_j' with clock condition in the form:

1. $\delta_i \vee \delta_j$, when at least one condition δ_i or δ_j shows fault propagation that results in a representation of the enabling time interval $(\delta_i \cup \delta_j) \setminus (\delta_i \cap \delta_j)$ which is distinguished in the specification. The fault propagation for δ_i and δ_j is given as in Definitions 5 and 6

2. $\delta_i \wedge \delta_j$, when at least one condition δ_i or δ_j shows fault propagation that results in a representation of the enabling time interval $(\delta_i \cap \delta_j)$ which is distinguished from the specification. The fault propagation for δ_i and δ_j is given as in Definitions 5 and 6.

6. CONCLUDING REMARKS

Model-based testing approaches for real-time systems have been explored in several works. The continuous time evolution and the reactive aspects of such systems are captured by TIOA models. Therefore a coverage analysis and a fault detection over these models is an important testing task aiding a fault model on identifying potential faults in real-time systems. However, continuous time evolution is a challenge to deal with, specially, in practical applications. Discretized models are then used to accomplish an analysis of fault detection for systems of this nature. Some works have been proposed in that direction, as the fault model for TIOAs based on region automata. But the coverage analysis based on region automata is restricted since the resulting discretization is tightly related to the number of clocks in the TIOA. The discretization approach in that work then incurs in the well-known state space explosion problem. Other discretizations for TIOA models have been proposed based on grid automata rather than region automata. Regarding these approaches, grid automatons are derived from TIOA models using an ample range of choice of granularities giving rise to a more manageable state space over discretizations. But the effectiveness of the fault detection in this approach falls short in specific faults using the notion of test purpose.

In this work, we proposed a fault model to characterize classes of faults in a TIOA through grid models. These faults have been classified into action faults, transfer faults, restriction and widening faults of clock condition, and faults of clock reset. We note that this fault model opens the possibility to generalize the fault detection process for real-time systems modeled by TIOAs.

As for future works we intend to perform experiments to show the fault model in practical applications. We also suggest the development of a test generation method based on this fault model. Finally we expect that our work provides the foundations for further proposals focused on fault coverage analysis and test suite extraction.

REFERENCES

[1] AR Dill DL. A theory of timed automata. Theor. Comput Sci 1994; 126: 183-235 http://dx.doi.org/10.1016/0304-3975(94)90010-8

[2] Alur R. Timed automata. In: Proceedings of the 11th International Conference on Computer Aided Verification. Springer-Verlag, London, UK (1999); 99: 8-22. http://dx.doi.org/10.1007/3-540-48683-6_3

[3] Blackburn M, Busser R and Nauman A. Why model-based test automation is different and what you should know to get started. In: International Conference on Practical Software Quality and Testing. PSQT/PSTT'2004 East, Washington, DC, USA (2004)

[4] Bonifacio AL and Moura AV. A new method for testing timed systems. Softw. Test, Verif Reliab 2013; 23(2): 91-117. http://dx.doi.org/10.1002/stvr.454

[5] Cardell-oliver R. Conformance testing of real-time systems with timed automata specifications. Formal Aspects of Computing 2000; 12(5): 350-371. http://dx.doi.org/10.1007/s001650070009

[6] Chow TS. Testing software design modeled by finite-state machines. IEEE Trans Softw Eng 1978; 4(3): 178-187. http://dx.doi.org/10.1109/TSE.1978.231496

[7] Clarke EM, Klieber W, Novácek M and Zuliani P. Model checking and the state explosion problem. In: Tools for Practical Software Verification, LASER, International Summer School 2011, Elba Island, Italy, Revised Tutorial Lectures. 2011; 1-30.

[8] En-Nouaary A. A scalable method for testing real-time systems. Software Quality Control 2008; 16: 3-22. http://dx.doi.org/10.1007/s11219-007-9021-8

[9] En-Nouaary A. Dssouli R and Khendek F. Timed wp-method: Testing real-time systems. IEEE Trans Softw Eng 2002; 28: 1023-1038 (November), http://portal.acm.org/ citation.cfm?id=630831.631295

[10] En-Nouaary A and Hamou-Lhadj A. A boundary checking technique for testing real-time systems modeled as timed input output automata (short paper). In: Quality Software, 2008. QSIC '08. The Eighth International Conference on 2008; 209-215.

[11] En-Nouaary A, Khendek F and Dssouli R. Fault coverage in testing real-time systems. In: Real-Time Computing Systems and Applications, 1999. RTCSA '99. Sixth International

Conference on. 1999; 150-157. http://dx.doi.org/10.1109/rtcsa.1999.811206

[12] Fujiwara S, von Bochmann G, Khendek F, Amalou M and Ghedamsi A. Test selection based on finite state models. IEEE Trans. Softw. Eng 1991; 17: 591-603. http://portal.acm.org/citation.cfm?id=126218.126234 http://dx.doi.org/10.1109/32.87284

[13] Gill A. Introduction to the theory of finite-state machines. McGraw-Hill electronic sciences series, McGraw-Hill (1962), http://books.google.com.br/books?id=lDhSAAAAMAAJ

[14] Hierons R, Bogdanov K, Bowen J, Cleaveland R, Derrick J, Dick J et al. Using formal specifications to support testing. ACM Comput. Surv 2009; 41(2): 1-76. http://dx.doi.org/10.1145/1459352.1459354

[15] Huth M and Ryan M. Logic in Computer Science: Modelling and Reasoning About Systems. Cambridge University Press, New York, NY, USA (2004) http://dx.doi.org/10.1017/CBO9780511810275

[16] Knight JC. Safety critical systems: Challenges and directions. In: Proceedings of the 24th International Conference on Software Engineering 2002; 547-550. ICSE '02, ACM, New York, NY, USA .

[17] Krichen M, Tripakis S. Black-box conformance testing for real-time systems. In: Model Checking Software: 11th International SPIN Workshop. pp. 109-126. No. 2989 in Lecture Notes in Computer Science, Barcelona, Spain (2004) http://dx.doi.org/10.1007/978-3-540-24732-6_8

[18] Springintveld J, Vaandrager F and D'Argenio PR. Testing timed automata. Theor. Comput. Sci 2001; 254: 225-257. http://dx.doi.org/10.1016/S0304-3975(99)00134-6

[19] Utting M and Legeard B. Practical Model-Based Testing: A Tools Approach. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2007)

[20] Wagner F. Modeling Software With Finite State Machines: Practical Approach. Modeling Software with Finite State Machines: A Practical Approach, CRC PressINC (2006)

Received on 11-12-2015 Accepted on 27-01-2016 Published on 14-06-2016

http://dx.doi.org/10.15379/2410-2938.2016.03.01.01