Poster Abstract: REVERT: Runtime Verification for Real-Time Systems

Sangeeth Kochanthara
IIIT-Delhi,
India

Geoffrey Nelissen
CISTER/INESC TEC, ISEP,
Portugal

David Pereira
CISTER/INESC TEC, ISEP,
Portugal

Rahul Purandare
IIIT-Delhi,
India

I. INTRODUCTION

Runtime verification (RV) is a formal technique that verifies properties during system execution with the support of monitors. The monitors are generated from formal languages using correct-by-construction generation methods. RV can thus be used as a complement or replacement for static verification approaches. The current state-of-the-art of these tools for real-time systems either do not have notion of time, or suffer from the potential blowup of states at run-time. In this paper, we propose REVERT, a framework developed with a focus on the verification of functional and non-functional properties with timing constraints. The contribution of this work is twofold: (i) a domain-specific specification language allowing the definition of requirements for real-time applications; (ii) a novel mechanism to generate monitors, with state-space and time guarantees, capable of identifying and reacting to timing properties defined with the proposed specification language.

II. SPECIFICATION LANGUAGE

REVERT is a new specification language for real-time applications designed with usability in mind. REVERT is a combination of state machine, extended regular expressions, boolean expressions, and timing constraints. REVERT relies on external events to reason about traces. Properties on execution patterns that must be enforced during the application run-time are specified using extended regular expressions.

To express timing constraints on a sequence of events we use three high-level operators: `time`, `duration` and `jitter` which return the time taken by a sequence of events, the execution time of a job, and the jitter on a timing property, respectively. These operators are then automatically converted to finite timed automata. The syntactic structure of a monitor specification in REVERT is presented below:

```
monitor m_i {
  observe { ev_1, ..., ev_l }
  variables {v_1 : type, ..., v_j : type }
  jobs{
    j_1{
     start: {ev_1, ev_2}
     suspend: {ev_3}
     resume: {ev_1}
     complete: {ev_6}
    },
    ...,
    j_p{...}
  }

  nodes { n_1, ..., n_k }
  initial { n_q }

  node n_1 { init_1 prop_1 trans_1 }
  ...
  node n_k { init_k prop_k trans_k }
}
```

The nodes model the different states that can be reached by a finite state machine. The transitions $trans_k$ between the nodes are triggered by guards based on the success or failure of the properties $prop_k$ monitored in that node n_k. For example, `failure`(`duration`(j_1) $<$ 10) triggers a transition if the execution time of the job j_1 is not smaller than 10 time units.

III. MONITOR GENERATION

In order to generate the monitor that will be running beside the application, we transform the specification to a complete deterministic finite automaton with the notion of time. This enables to generate a monitor with time and space guarantees by ensuring the monitor is tracking a single state at any time. To the best of our knowledge no other RV tool with the notion of time gives state-space and time guarantees.

The generation of monitors is achieved through the following steps: 1) Generating an automaton for each transition; 2) Generating an automaton for each node by applying a product operation on the automata obtained for each transition from the current node to any other node. Implicit priority is used to resolve potential conflicts on the final state; 3) Generating the monitor automaton by concatenating the automata of all nodes.

IV. FUTURE WORK

As future work, we first plan to formally prove the correctness of the algorithm to generate timed DFA from the properties specified using the `time` operator. Secondly, we will compare the expressivity of our language with state-of-the-art tools. Finally, we will bound the time and space complexity of the generated monitors.

ACKNOWLEDGMENTS

This work was partially supported by National Funds through FCT/MEC (Portuguese Foundation for Science and Technology) and co-financed by ERDF (European Regional Development Fund) under the PT2020 Partnership, within the CISTER Research Unit (CEC/04234); also by ARTEMIS/0001/2013 - JU grant nr. 621429 (EMC2).