Run Time Monitoring: A Design Perspective

A. Prasad Sistla and Miloš Žefran
University of Illinois at Chicago, Chicago, IL
{sistla,mzefran}@uic.edu

The growing complexity of modern engineered systems, and their increased reliance on computation, calls for novel approaches to guaranteeing their correct functioning. This is especially important for automotive systems where a failure can have catastrophic consequences.

One way to ensure correctness of a complex system is to thoroughly test and/or verify it. While testing can increase confidence in a component, it can not guarantee correctness. Verification, on the other hand, can guarantee correctness, but it is simply not feasible, for example, for a car with advanced engine controls and numerous networked microprocessors. In other cases, the component might have been verified for correctness on a model which was not accurate. And more importantly, even if a component is found to be defective through verification, we may still want to use it if the incorrect behavior only occurs rarely.

Run time monitoring of the behavior of a component is an approach that can complement testing and verification. It can provide another layer of safety to the operation of the system. The monitor observes the inputs and outputs of the component and checks whether the behavior of the system is consistent with the expected behavior. Monitors can be especially useful if a fail-safe shutdown procedures can be developed, which is true for a broad class of systems. We propose that monitor design be separate from the system design and be performed after the design of the system by a different set of designers. The fundamental advantage of monitors is that they are in principle easy to design and implement, and they do not fundamentally constrain the design of a component. Such two layer approach ensures that incorrect behaviors, due to potential faulty component designs, are detected by the monitor and are acted upon.

1 Formalism Description

This section briefly summarizes our existing work on run time monitoring for cyber-physical systems [3]. We assume that the system behavior is stochastic and the state of the system is not directly observable. Furthermore, since we are motivated by electronic control systems, we assume that the state of the system is quantized. We consider Hidden Markov Chains (HMC) to model such discrete state systems. HMCs can have countably infinite number of states and can thus model a wide range of engineering systems. In particular, together with their extensions, they are well suited to model non-determinism arising from integration of independently-designed deterministic components.

We assume that the property to be monitored is specified by an automaton $\mathcal{A}$ on infinite trajectories/computations of system states. A monitor of the system, given by a HMC H, observes the system outputs and raises an alarm whenever it determines that the system computation is wrong, i.e., does not satisfy the property specified by $\mathcal{A}$. To characterize the effectiveness of a monitor, we define two accuracy measures, called *Acceptance Accuracy* (AA) and *Rejection Accuracy* (RA). Considering raising of an alarm as rejection, and not

raising an alarm as acceptance, AA denotes the percentage of good computations of the system that are accepted by the monitor. RA denotes the percentage of bad computations that are rejected by the monitor. Ideally, both these accuracies should be equal to 1. The values $(1 - AA)$ and $(1 - RA)$ are measures of false alarms and missed alarms, respectively, and should be kept low.

In [3] we propose two notions of monitorability of a system given as a HMC H with respect to a property automaton $\mathcal{A}$. H is said to be *strongly monitorable* with respect to $\mathcal{A}$ if there is a monitor for which both accuracies have values 1. Since strong monitorability is rarely achieved, we define a weaker notion, *monitrability*. We say that H is *monitorable* with respect to $\mathcal{A}$ if accuracies arbitrarily close to 1, can be achieved; that is for every $x \in [0, 1)$, there is a monitor such that both of its accuracies are greater than or equal to x. Note that monitorability is different from the classical notion of observability in systems theory. Monitorability depends on the system as well as the property, while observability is an inherent property of the system. Observability of a system implies that it is monitorable with respect to every property. On the other hand, one can easily construct realistic systems that are not observable but are monitorable with respect to a property.

2 Research Agenda

Run time monitoring constitutes an important tool for design of dependable electronic control systems, but many important research problems remain open. We thus propose the following research agenda.

Theories of Monitorability: The above definitions of monitorability do not consider the time taken to raise an alarm after the system computation becomes bad. So, it is necessary to develop alternate definitions of monitorability and extend them to the case when some of the system states are continuous. In each of the cases, it would be desirable to exactly characterize when a system is monitorable with respect to a property. We have already obtained such results (see [3]) for monitorability and strong monitorability.

Formal Models and Languages: It is necessary to develop languages and formalisms for specifying system behaviors and properties to be monitored. These should be tailored to a specific application area and be easy to use by practitioners. They can be derived from the existing well known formalisms such as Probabilistic Hybrid Automata [2] for describing systems and Deterministic Hybrid Automata [1] for specifying properties.

Cost Based Monitoring: When available resources (e.g., time and computation budget) are limited, the accuracy measures proposed above, do not sufficiently capture the design trade offs that need to be made. For example, a missed alarm could have dramatically different impact in different situations. Similarly, an alarm that is raised too late is useless. We call for investigation of different cost models and different techniques (e.g., Dynamic Bayesian networks or Partially Observable Markov Decisions Processes) for the design of optimal monitors under these cost models.

Reconfigurable and Adaptive Systems: When a system model is only partially known, or a component design is proprietary, it is necessary to develop techniques that can achieve monitorability despite incomplete knowledge of the system. Adaptation and learning are important strategies that need to be explored for monitoring of such systems.

Hierarchical and Distributed Architectures: As complex systems are assembled from components developed by various suppliers, possibly with their embedded monitors, it is necessary to develop methodology for composing monitors of individual components so that the overall system behavior can be monitored. We propose to use both hierarchical and distributed architectures for composing monitors. More generally, formal approaches for composing individual monitors should also be explored.

Standardization and Regulatory Issues: Since run time monitors can be designed separately from the rest of the system, they can provide an additional layer of assurance at modest investment. We propose that, for safety critical systems, run time monitoring should become an integral part of a sound design process. This should be investigated in the context of standardization and regulatory activities.

About the Authors

A. Prasad Sistla obtained Ph.D. degree in Computer Science/Applied Mathematics from Harvard University in 1983. Prior to that he obtained M.E. degree in Computer Science from Indian Institute of Science, Bangalore, India. He is currently a Professor in the Department of Computer Science in the University of Illinois at Chicago. Prof. Sistla has done extensive research in the areas of Model Checking, Analysis and Verification of Concurrent Systems, Formal Methods for Concurrent and Distributed Systems and also in Mobile Database Systems. He was a co-chair of the 2000 International Conference on Computer Aided Verification and has been funded by leading NSF, AFOSR and DARPA. He was also a consultant for Bell Labs/Lucent Technologies and Microsoft Research.

Miloš Žefran completed his undergraduate education at the University of Ljubljana, Slovenia, and received a Ph.D. degree in Computer Science from the University of Pennsylvania in 1996. He was a NSF Postdoctoral Scholar at the California Institute of Technology. Since 1999 he has been with the Department of Electrical and Computer Engineering at the University of Illinois at Chicago where he is currently an associate professor. His research interests are in robotics and control. He received the NSF Career Award in 2000 and is an Associate Editor of the IEEE Transactions on Control Systems Technology.

References

[1] R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R. Grossman, A. Nerode, A. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer Berlin / Heidelberg, 1993.

[2] M. W. Hofbaur. Probabilistic hybrid automata. In M. W. Hofbaur, editor, *Hybrid Estimation of Complex Systems*, volume 319 of *LNCS*, pages 29–43. Springer, 2005.

[3] A. P. Sistla, M. Žefran, and Y. Feng. Monitorability of stochastic dynamical systems. Technical Report CVRL-2011-01, Computer Vision and Robotics Laboratory, University of Illinois at Chicago, Chicago, IL, 2011. `http://www.cvrl.cs.uic.edu/~milos/publications/papers/cav2011_long.pdf`.