

Stream Runtime Verification of Uncertain Traces

Motivation and Background

Different verification techniques are used throughout the engineering process of systems. For many systems, this process of verification must not end with the system's deployment, but needs to continue at runtime. One technique to provide formal assurance at runtime is stream runtime verification (SRV) [1]. In SRV, a stream of system events (e.g., from a log or from message exchange in a distributed system) is analysed w.r.t. a requirement expressed in a formal language such as Linear Temporal Logic (LTL). This formal requirement (a so called property) can automatically be transformed into an automaton [2] to check if a system execution conforms to the requirement, i.e., solve the word problem (is the execution (word) successful or failing (going to good states or bad states)). Most approaches, however, assume perfect knowledge about the observed system. In other words, they assume that the trace of events is known perfectly. However, in many systems, such as distributed or cyber-physical systems, this assumption might not hold. In previous, unpublished work we have developed an initial idea of runtime verification under uncertain traces.

Goals

The goal of this thesis is to extend this initial idea of runtime verification under uncertainty and apply it to mature systems to evaluate the (dis)advantages of this technique.

Description of the Task

The specific tasks are:

- Find related approaches in the literature
- Evaluate our existing implementation

Research Type

Theoretical Aspects:	*****
Industrial Relevance:	*****
Implementation	*****

Prerequisite

The student should be enrolled in the bachelor or master of computer science program, and has completed the required course modules to start a bachelor or master thesis.

Skills required

Programming skills in C++, understanding of, or willingness to learn, formal methods such as needed for the project. The student should be able to read and comprehend research articles in English.

Contacts

Marc Carwehl (m.carwehl@hu-berlin.de)

Software Engineering Group, Institut für Informatik, Humboldt-Universität zu Berlin

References

[1] Falcone, Yliès, Klaus Havelund, and Giles Reger. "A tutorial on runtime verification." *Engineering dependable software systems*(2013): 141-175.

[2] Havelund, Klaus, and Grigore Roşu. "Synthesizing monitors for safety properties." *Tools and Algorithms for the Construction and Analysis of Systems: 8th International Conference, TACAS 2002 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2002 Grenoble, France, April 8–12, 2002 Proceedings 8*. Springer Berlin Heidelberg, 2002.