Machine Learning for Continuously Improving Safety-Critical Autonomous Systems

Piergiuseppe Mallozzi
Chalmers University of Technology

Description:

Autonomous systems are self-adaptive systems that are able to adapt their behaviour at run-time without human intervention in response to changes in the environment or in their internal state. As the environment produces new data, machine learning techniques can use this data to drive the system adaptation. Unlike traditional software it is much harder to verify that machine learning methods produce solutions that are compliant with the systems requirements and in safety-critical system this could lead to hazardous situations. We use predictive monitoring to continuously check at run-time that safety-critical requirements are respected and by doing so we train the machine learning component so that it can learn to produce better solution in the future.

Background & Motivation

Despite the potential drawbacks of using machine learning in safety-critical systems, such techniques can bring many advantages as well, especially in Self-Adaptive Systems. Because the environment continuously changes producing new data, traditional software must be updated so it can produce better solution every time. Instead, using machine-learning to drive the self-adaptation we can achieve a continuous learning from the environment as new data comes in the machine-learning models. Furthermore, in an environment with a lot of variability and uncertainty, it's often impossible to define every behaviour that the system should have. By using machine learning we can set the goals we want to achieve and let the system figure out how to reach those goals by itself.

Research Aims

Our research aims to combine machine learning approaches with assurances of the autonomous systems safety-critical requirements. We will focus on the integration of different state-of-the-art methods with the vision to build an autonomous system that can make safety-certifiable decisions.
Our goal is to model a system that continuously evolves using machine learning techniques to drive the system's adaptations. At the same time, we want to keep the preservation of the system's invariants by continuously monitoring how the system's react to the changes of the environment.

Methods & Preliminary Results

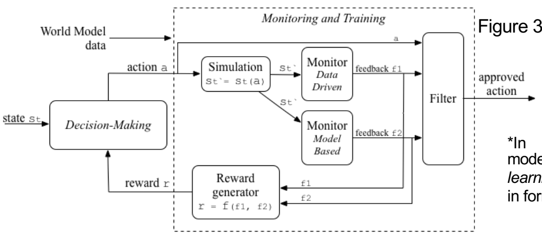

Figure 3. Two-Layer Monitoring

*In a decision-making agent modeled with *reinforcement learning* the feedback is generate in form of *reward*

We present a method summarized with the feedback loop shown in Figure 1. The decision-making component is composed of agents that produce actions to achieve the goals set by the World Model component. Each action is checked to prevent the system from reaching wrong states with two layer monitoring described in the Figure 3. We use predictive monitoring to prevent dangerous actions from being executed and to train the decision-making component so that it can continuously improve*.

Roadmap & Milestones

- Analyzed of today's safety standards such as ISO26262 for the automotive industry
- Applied formal methods to verifying vehicle platooning protocol [1], we've used model checking to verify safety invariants. Easily scalable with the number of vehicles and automatic creation and verification of multiple scenarios. However everything has to be checked at *design-time*.
- Defining a method that combines machine learning and continuous monitoring of the system's invariants at *run-time* [2].
- Experimenting it in the automotive domain with different levels of adaptation (*self-adaptation* and *evolution*).

[1] Mallozzi Piergiuseppe, Massimo Sciancalepore, and Patrizio Pelliccione. "Formal Verification of the On-the-Fly Vehicle Platooning Protocol." *International Workshop on Software Engineering for Resilient Systems*. Springer International Publishing, 2016.

[2] P. Mallozzi "Combining Machine-Learning with Invariants Assurance Techniques for Autonomous Systems" submitted to ICSE Doctoral Symposium, 2017

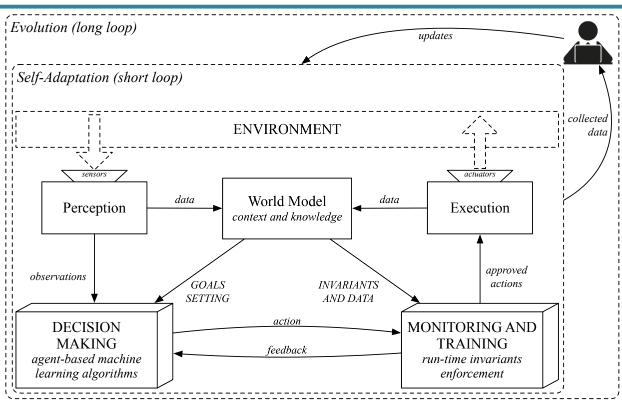

Figure 1. Proposed method

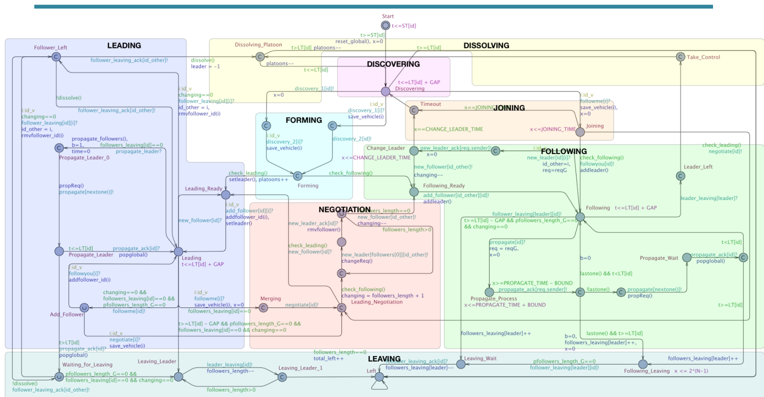

Figure 2. Model checking of vehicle platooning protocol using Uppaal

WALLENBERG AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM