Science of Computer Programming 148 (2017) 88–106

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Formal verification of autonomous vehicle platooning

Maryam Kamali [a,*], Louise A. Dennis [a], Owen McAree [b], Michael Fisher [a], Sandor M. Veres [b]

[a] Department of Computer Science, University of Liverpool, UK
[b] Department of Automatic Control & Systems Engineering, University of Sheffield, UK

ARTICLE INFO

Article history:
Received 6 June 2016
Received in revised form 25 May 2017
Accepted 30 May 2017
Available online 14 June 2017

Keywords:
Vehicle platooning
Agent programming
Model checking

ABSTRACT

The coordination of multiple autonomous vehicles into convoys or platoons is expected on our highways in the near future. However, before such platoons can be deployed, the behaviours of the vehicles in these platoons must be certified. This is non-trivial and goes beyond current certification requirements, for human-controlled vehicles, in that these vehicles can act *autonomously*. In this paper, we show how formal verification can contribute to the analysis of these new, and increasingly autonomous, systems. An appropriate overall representation for vehicle platooning is as a multi-agent system in which each agent captures the "autonomous decisions" carried out by each vehicle. In order to ensure that these autonomous decision-making agents in vehicle platoons never violate safety requirements, we use formal verification. However, as the formal verification technique used to verify the individual agent's code does not scale to the full system, and as the global system verification technique does not capture the essential verification of autonomous behaviour, we use a combination of the two approaches. This mixed strategy allows us to verify safety requirements not only of a model of the system, but of the actual agent code used to program the autonomous vehicles.

1. Introduction

While "driverless cars" regularly appear in the media, they are neither "driverless" nor fully autonomous. Legal constraints, such as the Vienna Convention [37], ensure that there must always be a responsible human in the vehicle. Although fully autonomous road vehicles remain futuristic, the automotive industry is working on what are variously called *road trains*, *car convoys*, or *vehicle platoons*. Here, each vehicle autonomously follows the one in front of it, with the lead vehicle in the platoon/convoy/train being controlled by a human driver. This technology is being introduced by the automotive industry in order to improve both the safety and efficiency of vehicles on very congested roads [29]. It is especially useful if the vehicles are trucks/lorries and if the road is a multi-lane highway.

In these platoons, each vehicle clearly needs to communicate with others, at least with the ones immediately in front and immediately behind. Vehicle-to-vehicle (V2V) communication is used at a lower (continuous control system) level to adjust each vehicle's position in the lanes and the spacing between the vehicles. V2V is also used at higher levels, for example to communicate joining requests, leaving requests, or commands dissolving the platoon. So a traditional approach is to implement the software for each vehicle in terms of hybrid (and hierarchical) control systems and to analyse this using hybrid systems techniques [4].

[*] Corresponding author.
E-mail address: maryam.kamali@liverpool.ac.uk (M. Kamali).

http://dx.doi.org/10.1016/j.scico.2017.05.006

However, as these automotive platoons become more complex, there is a move towards much greater autonomy within each vehicle. Although the human in the vehicle is still responsible, the autonomous control deals with much of the complex negotiation to allow other vehicles to leave and join, etc. Safety certification is an inevitable concern in the development of more autonomous road vehicles, and verifying the safety and reliability of automotive platooning is currently one of the main challenges faced by the automotive industry. Traditional approaches to modelling such situations involve hybrid automata [18] in which the continuous aspects are encapsulated within discrete states, while discrete behaviours are expressed as transitions between these states. A drawback of the hybrid automaton approach is that it can be difficult to separate the two (high-level decision-making and continuous control) concerns. In addition, the representation of autonomous decision-making can become unnecessarily opaque in such hybrid approaches.

As is increasingly common within autonomous systems, we use a hybrid architecture where not only is the discrete decision-making component separated from the continuous control system, but the behaviour of the discrete part is described in much more detail; in particular, using the *agent* paradigm [40]. This style of architecture improves the system design from an engineering perspective and also facilitates system analysis and verification [9]. Indeed, we use this architecture for actually implementing automotive platoons, and we will here show how formal verification can be used for its direct analysis.

The verification of such systems is challenging due to their complex and hybrid nature. Separating discrete and continuous concerns, as above, potentially allows us to reason about the decision-making components in isolation and ensure that no decision-making component ever deliberately chooses an unsafe state. However, the use of the ‘agent’ concept alone is not enough for our purposes, since this can still make its autonomous decisions in an ‘opaque’ way. In order to be able to reason about, and formally verify, the choices the system makes, we use a *rational agent* [41]. This not only makes decisions, but has explicit representations of the *reasons* for making them, allowing us to describe not only what the autonomous system chooses to do, but *why* it makes its particular choices [16].

We utilise the *Belief-Desire-Intention* (BDI) model, one of the most widely used conceptual models, not only for describing these rational agents but for actually implementing them [32]. A BDI-style agent is characterised by its beliefs, desires and intentions: *beliefs* represent the agent’s views about the world; *desires* represent the objectives to be accomplished; while *intentions* are the set of tasks currently undertaken by the agent to achieve its desires. A BDI-style agent has a set of plans, determining how an agent acts based on its beliefs and goals, and an event queue where events (perceptions from the environment and internal subgoals) are stored. There are several advantages in using this style of model for developing autonomous systems: (a) it naturally separates feedback controllers from high-level decision making, as above; (b) it facilitates reasoning and verifying about the behaviour of high-level decision making [11]; (c) it supports incremental and hierarchical development of plans; and (d) it provides a clear separation between plan selection and plan execution. However, a drawback of this form of approach is that it is essentially a plan management and plan selection framework with no in-built mechanisms for learning or first-principles planning. This means that a BDI agent does not automatically learn from past behaviour and adapt its plans accordingly. A similar limitation is lack of predictability and forward planning, in its basic form. As the aim of this paper is on *verification* of *decisions* concerning platooning, we can utilise this model in the form of the Gwendolen programming language [8], developed for verifiable BDI-style programming, to capture and implement the agent-based decision-making in each vehicle within an automotive platoon.

As part of safety verification, we need to verify the agent’s decisions, especially in combination with the other vehicles. An autonomous rational agent makes decisions about what actions to perform, etc., based on the *beliefs*, *goals* and *intentions* that the agent holds at that time. We use a model-checking approach to demonstrate that the rational agent always behaves in line with the platoon requirements and never deliberately chooses options that end up in unsafe states. We verify properties of the rational agent code using the AJPF model-checker [12], one of the very few model-checkers able to cope with complex properties of BDI agents. Unfortunately, there are two drawbacks to using AJPF: currently, AJPF does not support verification of timed behaviours; and AJPF is resource heavy and cannot be used to verify the whole system. Consequently, we here propose a combined methodology for the verification of automotive platooning. To evaluate timing behaviour, we use a timed-automata abstraction and verify the system using Uppaal [2]; to evaluate individual autonomous decisions, we use AJPF together with an abstraction of the other vehicles/agents. Furthermore, we describe how these two approaches to modelling can be combined to provide an appropriate basis for verifying the behaviour of both individual agents and the whole system.

Overview of our contribution. The ISO 26262 standard provides a standard for *functional safety* management in automotive applications, and determines the safety requirements that should be fulfilled in design, development and validation of individual automotive units. Following the ISO 26262 guidelines for safety management in automotive applications we propose a verifiable agent-based architecture for development of safe automotive platooning and, in the same research line, we propose new combined verification techniques for autonomous systems developed based on hybrid agent architectures. In particular, we show the applicability of our verification techniques in the development of platooning.

For a clear picture of our approach, we summarise our fourfold contribution.

I We introduce formal *automotive platoons requirements*. This allows us to better understand the functioning of platooning protocols and, more importantly, to verify essential properties such as the functional correctness and liveness of the protocols. An important aspect of the protocols is that a vehicle can join and leave a platoon if, and only if, the whole platooning remains safe.