Formalisation of the Rules of the Road for embedding into an Autonomous Vehicle Agent

Gleifer Vaz Alves[1], Louise Dennis[2] and Michael Fisher[2]

[1]*UTFPR - Federal University of Technology - Paraná - Brazil*
[2]*Department of Computer Science, Univ. Liverpool, UK*

Abstract

We propose a description language for formalising the "Rules of the Road" (e.g., the UK Highway Code). By way of example, we represent the subset of the Highway Code responsible for road junctions and represent these Rules of the Road using linear temporal logic, LTL. Our aim is to "extract" from this generic representation a set of beliefs, goals and plans that can be used by a BDI agent to augment its universal autonomous vehicle control (e.g., route planning, and obstacle avoidance capabilities) with region-specific rules for road use. We intend to target the GWENDOLEN agent programming language as a prototype for this system, which provides formal verification possibilities.

Introduction

Research into the development of autonomous vehicles is popular, including research into the rational agent-based control of these vehicles. We focus here on one aspect of this: are these agent-based autonomous systems ready to adopt different kinds of highway code? For instance, imagine that an autonomous vehicle is sold in the UK and configured to drive on UK roads, however, if it crosses the channel it will be suddenly driving on French roads. If we have a rational agent controlling the car, it would be desirable if this agent could easily adopt a new set of rules according to the local Rules of the Road. With this in mind, we aim to create a simple formal language in which region-specific Rules of the Road can be described. This language uses temporal logic operators and actions. As an initial example, we consider a subset of the Rules of the Road for a single region: the road junction rules for the UK Highway code [3].

Our aim is to translate the simple description into beliefs, goals and plans in the GWENDOLEN rational agent programming language [1]. In particular we aim to show how these new components can integrate with an existing GWENDOLEN program for generic Autonomous Vehicle (AV) control. That is, given a GWENDOLEN agent that represents the behaviour of an Autonomous Vehicle (AV) and consequently has beliefs, goals and plans which have been (previously) designed and verified for driving, avoiding obstacles, detecting pedestrians and traffic signs and so on, we intend to add an additional set of beliefs, goals and plans to the AV's agent. With this new set ("extracted" from the Rules of the Road), the AV agent is not only able to drive, avoid obstacles, etc but also can also follow the guidelines that have been described in the Rules of the Road language.

Formalisation of the Rules of the Road

In the UK Highway code there are 14 rules which deal with road junctions. These cover behaviour for junctions, box junctions, dual carriageways, and so on. For example, rule **170** describes how a driver should (safely) enter a road junction. One of its conditions says: "you should watch out for pedestrians crossing the road".

Rule **170** is formalised with in our language using LTL operators [5], objects and actions. The objects are *Pedestrian* (i.e. a `road-user`) and `junction`, while the action is `watch`. Moreover, we have an intention to: "safely enter a junction".

This rule fragment is represented in our language as:

$$\Box\ (\texttt{watch(AV,junction,road-user)} \cup \texttt{check(cross(road-user,junction))=False}) \rightarrow \Diamond\ \texttt{(enter(AV,junction))}$$

That is, *it is **always** the case the AV should **watch out** for road users at a road junction **until** there is no road user crossing the junction. Then, the AV will **eventually** enter the junction.* From this type of formula, we will extract: **i.** two new Plans with the actions `watch` and `check-for-crossing`, which are applicable when there is a Belief that there exists a `road-user` at the `junction`; and **ii.** a new Goal to "safely **enter** a junction", which will be triggered by a new Belief (i.e., there is no `road-user`). These plans should be passed to the GWENDOLEN agent for use.

Final Remarks

In previous work [4], we have implemented and verified some properties concerning basic driving capabilities of a rational agent represented as an AV. In [7] the "rules of the air" were defined in a GWENDOLEN agent and also formally verified. Moreover, the authors in [6] have formalised traffic rules using Isabelle theorem prover. In our approach we intend to formalise the rules of the road separately from the GWENDOLEN agent and consequently devise a system capable of adopting region-specific rules.

Furthermore, once we have incorporated our new beliefs, goals and plans into a GWENDOLEN agent we aim to use model checking techniques via the MCAPL (*Model Checking Agent Programming Languages*) framework [2], in order to verify whether the modified AV agent will actually follow the guidelines from the Rules of the Road. The MCAPL framework has a property specification language based on LTL and so our intention is to use the rules expressed in our abstract language directly as a specification and then verify that the extracted GWENDOLEN plans, combined with the agent's pre-existing plans, are correct.

As further work we intend to consider a larger set of rules from the UK Highway code and include other regional Highway codes, e.g. Portuguese, and Brazilian codes.

Acknowledgements

Work partially supported through EPSRC research project Verifiable Autonomy [EP/L024845].

References

[1] L. A. Dennis. Gwendolen semantics: 2017. Technical Report ULCS-17-001, University of Liverpool, Department of Computer Science, 2017.

[2] L. A. Dennis, M. Fisher, M. P. Webster, and R. H. Bordini. Model Checking Agent Programming Languages. *Automated Software Engineering*, 19(1):5–63, Mar. 2012.

[3] Department for Transport. Using the road (159 to 203) - The Highway Code - Guidance - GOV.UK, 2017.

[4] L. E. R. Fernandes, V. Custodio, G. V. Alves, and M. Fisher. A Rational Agent Controlling an Autonomous Vehicle: Implementation and Formal Verification. In L. Bulwahn, M. Kamali, and S. Linker, editors, *Proceedings First Workshop on Formal Verification of Autonomous Vehicles*, volume 257 of *Electronic Proceedings in Theoretical Computer Science*, pages 35–42, 2017.

[5] M. Fisher. *An Introduction to Practical Formal Methods Using Temporal Logic*. Wiley, 2011.

[6] A. Rizaldi, J. Keinholz, M. Huber, J. Feldle, F. Immler, M. Althoff, E. Hilgendorf, and T. Nipkow. Formalising and monitoring traffic rules for autonomous vehicles in isabelle/hol. In *Proc. 13th International Conference on Integrated Formal Methods*, pages 50–66, 2017.

[7] M. Webster, N. Cameron, M. Fisher, and M. Jump. Generating Certification Evidence for Autonomous Unmanned Aircraft Using Model Checking and Simulation. *Journal of Aerospace Information Systems*, 11(5):258–279, 2014.