Towards a verifiable decision making framework for self-driving vehicles

Mohammed Al-Nuaimi[1], Hongyang Qu[2] and Sandor M. Veres[3]

***Abstract*— A new verification framework is presented for the decision making of autonomous vehicles (AVs). The overall structure of the framework consists of: (1) A perception system of sensors that feed into (2) a reasoning agent based on a Jason architecture that operates on-board an AV and interacts with a model of the environment using (3) multi-input multi-output adaptive control system. The agent relies on a set of rules, planners and actions to achieve its ultimate goal of driving the AV safely from a starting point to its destination. The verification framework deploys an innovative combination of two well known verification tools: (1) the model checker for multi-agent systems (MCMAS) to check the consistency and stability of the agent logic rules and perception-based beliefs before choosing any action and (2) the PRISM probabilistic model checker to provide the agent with the probability of success when it decides to take a planned movement sequence. The agent will select the movement-actions with the highest probability of success. The planned actions are executed by a control system to control the movement of the AV on the ground using a set of primitive movement skills using its actuators. The framework adopts the Jason agent paradigm to design the reasoning and the decision making process. The Robot Operating System (ROS) and the Gazebo Simulator are used to model the AV, its sensors and the surrounding environment. The agent connects to a MATLAB-based perception and control system to steer the AV.**

I. Introduction

It is a standard requirement that an AV should be able to progress without human control from its initial position to its final destination and handle various interruptions during its journey. This process involves making decisions by a multi-process software on-board the AV that observes the environment, applies a mission planner and uses an intelligent agent to coordinate these processes. Agents have been rapidly developed during the past two decades. Some notable agent types are reactive, deliberative, multi-layered, and belief-desire-intention (BDI) agents [1]. Jason is a multi-layered well-known approach to intelligent agents with BDI paradigm, which is particularly suitable for achieving goals [2] by an AV.

Reconfigurable, adaptive, and rational-agent based control-systems are well capable of robustly progressing a vehicle in space and time to avoid other vehicles and people [3] [4]. However, to make decisions with foresight, and consideration to other traffic participants in a social context, integration is essential into an overall decision-making that is based on behaviour rules and experience. Rational agents have significant potential in the implementation of the various applications. However, for real-world applications they will raise some safety concerns, creating the need for verification framework. Testing of these systems through prototype development will try to answer operational safety questions partially. The best can be done in testing is a representative set of scenarios on real vehicles. While simulations can only provide illustrations of correct social behaviour of the AV and cannot take into account a rare combination of events that may arise during run-time of the autonomous system and lead to the chosen action. Hence verification methods are needed.

If good dynamical models are available to encapsulate robotic skills of sensing and action, then formal verification can be used to verify properties of a finite representation of an interacting model of the agent and the robot's environment. The autonomous agent's decisions depend on the current state of the environment, and the way it perceives data from its sensors. Due to the uncertainty associated with the environment probabilistic model checking is needed to be used.

II. General Framework

The verification framework models an AV's decision system and implements it using ROS [5], part of which is interaction with a Gazebo-based virtual reality simulator [6] to model the vehicle dynamics, the sensors needed for perception and a model of the traffic environment. A MATLAB-based control system is used for path planning and to steer the vehicle. The agent has both rule-based reasoning and a set of predicates for perception and control events. The logic based reasoning system has been implemented by natural language programming (NLP) in sEnglish [7] that compiles into a Jason agent [8]. The perception system generates a model of the surrounding areas that the vehicle passes through with the ability to localize static and dynamic objects and keep updating its model.

Fig. 1 shows a simplified overall diagram of the proposed framework. A novelty of this framework is that the verification system is able to check for any instability and inconsistency using MCMAS [9], dependent of a logic-rule set. Following this, the various environmental events, and options for movements are modelled and verified during run-time using PRISM [10] through Probabilistic Time Program (PTP) to obtain a set of probabilities for the vehicle's actions to enable the agent to make reasoned choices.

[1]Mohammed Al-Nuaimi is a PhD student at Department of Automatic Control and Systems Engineering (ACSE), University of Sheffield. myhazim1@sheffield.ac.uk

[2]Dr Hongyang Qu is Senior Research Fellow at ACSE, University of Sheffield. h.qu@sheffield.ac.uk

[3]Sandor M. Veres is a Professor at ACSE, University of Sheffield. s.veres@sheffield.ac.uk

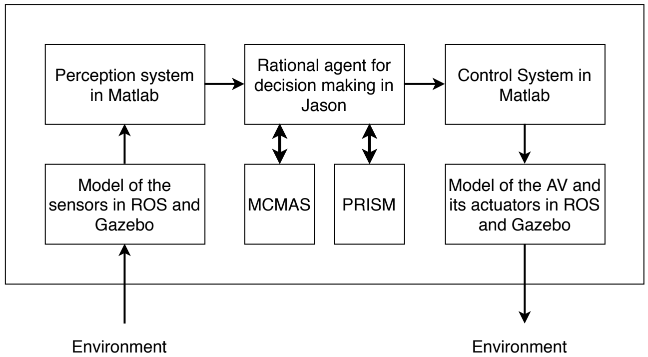

Fig. 1. General Framework of the project

III. Formal verification of decision making

The feasibility and logical consistency of run-time decisions by a rational agent is an important issue for the future of AVs and for robotics in general. The developed rule-set and planning based robotic system has a belief set as in the BDI approach and is the basis of its reasoning. The belief predicates of our AV are abstracted and derived from the sensors in the simulated or real environment. Three types of rules are used: physical rules, social behaviour rules, and consequence rules based on naive physics, and the robot has the freedom to choose an action that is compatible with all belief predicates. Inconsistent beliefs may lead to an unsafe action causing dangerous behaviour, when there is more than one possible action, while unstable rules will prevent the AV from selecting an action in a timely manner.

In this work we are interested in both design-time and run-time verification, this process involve the extraction and analysis of information from the system to detect possible react to observed behaviours satisfying or violating certain properties. Our verification engine consist of design-time verification using MCMAS and run-time verification using PRISM, a prototype of a translators are used to build the models for the model checkers.

MCMAS is used to check the consistency and stability of beliefs, rules, and actions of our AV in its environment. Both consistency and stability can be captured by CTL formulae and models checked efficiently by MCMAS [11]. When the system is not consistent or stable, a counterexample is generated to demonstrate the violation and help the developers correct the system.

PRISM is used by the agent at run-time to ask question like: what is the probability of success of the current action or what is the probability of achieving the current goal with in a time limit [10]. For example, agent can ask what is the maximum probability of success for the AV moving to a specific location within a specific time period taking into account other objects moving around. The agent program and its physical environment have been modelled as a Probabilistic Timed Program (PTP) in terms of the predicates fed back to the belief base of the agent under variant states of the environment while the Probabilistic Computational Tree Logic (PCTL) is used for specification logic. A PTP consists of the states of the environment and the transition between those states, which through the conditional probabilities of the environment correspond to triggering of predicates through the sensor system of the AV.

The rules are used to set the relationship between the perception predicates (beliefs) and the available actions, when this combination get verified by MCMAS then there will be no space for an infeasible action in the agent's actions list, this will be reflected later on the run-time verification by reducing the state-space available to check by PRISM. This process is another advantage of our proposed verification engine.

IV. Conclusion

This work has defined a new verification framework for agent-based decision-making of an AV by adopting some existing methods and developing some new ones. The general framework includes a configuration of sensors to sense the surrounding environment with a number of skills to control the movement of the AV. A social-rules based agent has been implemented to guide the AV through a set of actions.

The novelty of this work is that the agent's decisions are verified from different sides. First MCMAS checks the consistency of the agent's beliefs that will lead to specific action to make sure that no conflict will occur while driving. It is verified that the rules are stable in a way that they can proceed in a timely manner. PRISM is used to enable the agent to choose a movement that is safe and considerate under the current driving scenario.

References

[1] R. H. Bordini, M. Fisher, C. Pardavila, and M. Wooldridge, "Model checking agentspeak," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. ACM, 2003, pp. 409–416.

[2] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *The knowledge engineering review*, vol. 10, no. 02, pp. 115–152, 1995.

[3] L. A. Dennis, M. Fisher, N. K. Lincoln, A. Lisitsa, and S. M. Veres, "Practical verification of decision-making in agent-based autonomous systems," *Automated Software Engineering*, vol. 23, no. 3, pp. 305–359, 2016.

[4] M. Y. Hazim, H. Qu, and S. M. Veres, "Testing, verification and improvements of timeliness in ros processes," in *Conference Towards Autonomous Robotic Systems*. Springer, 2016, pp. 146–157.

[5] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, vol. 3, 2009.

[6] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2149–2154.

[7] S. M. Veres, "Natural language programming of agents and robotic devices," 2008.

[8] R. H. Bordini, J. F. Hübner, and M. Wooldridge, *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons, 2007, vol. 8.

[9] A. Lomuscio, H. Qu, and F. Raimondi, "Mcmas: A model checker for the verification of multi-agent systems," in *International Conference on Computer Aided Verification*. Springer, 2009, pp. 682–688.

[10] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *Computer aided verification*. Springer, 2011, pp. 585–591.

[11] H. Qu and S. M. Veres, "Verification of logical consistency in robotic reasoning," *Robotics and Autonomous Systems*, vol. 83, pp. 44–56, 2016.