Verifiably Safe Decision-Making for Autonomous Systems

Doctoral Consortium

Yi Yang
imec-DistriNet, KU Leuven
Leuven, Belgium
yi.yang@kuleuven.be

ABSTRACT

Autonomous systems have the potential to significantly boost the productivity of our society. However, safety concerns are the primary impediment to the widespread use of autonomous systems. Safe decision-making for autonomous systems is a crucial step toward developing safe autonomous systems. My Ph.D. topic focuses on a formal approach to efficiently generating verifiable safe decision-making for autonomous systems. I have designed and implemented a three-stage formal approach to addressing the issue, and I have validated my approach with a real-world autonomous logistic system consisting of three autonomous mobile robots. This paper summarizes my current work and outlines my future work.

KEYWORDS

Autonomous System; Decision-Making; Formal Specification; Interpreter; Probabilistic Model Checking

ACM Reference Format:
Yi Yang. 2023. Verifiably Safe Decision-Making for Autonomous Systems: Doctoral Consortium. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

Autonomous systems have the potential to significantly boost the productivity of our society, as they can make their own decisions without external instructions. However, safety concerns have led many to express skepticism about autonomous systems such as self-driving cars. Therefore, providing safety assurance for autonomous systems is vital, albeit challenging. Safe autonomous decision-making is a crucial research topic for ensuring the safety of autonomous systems.

[15] provides a high-level overview of an approach for formally verifying the behaviors of autonomous systems. Building on the approach described in [15], [12] presents a verification methodology for the decision-making component in agent-based hybrid systems. [14] introduces an architectural framework for developing verifiable self-certifying autonomous systems. [16] analyzes the key aspects to develop a framework for the certification of reliable autonomous systems. [10] presents a framework for verifiable autonomous decisions and its applications to assessing a range of properties of autonomous systems.

My research builds upon the assumptions and definitions established in the aforementioned research work [10, 12, 14–16]. Specifically, I focus on the design and analysis of a safe agent-based decision-making component that serves as a high-level discrete controller for an autonomous system. This component operates independently but collaboratively with the low-level continuous controller of an autonomous system [12]. Given the inherent unpredictability of the real world, it is impossible to guarantee that any system will always behave safely [10]. Consequently, I consider an autonomous decision-making component safe if it avoids deliberately pursuing unsafe behaviors based on its beliefs and goals [10]. Furthermore, I use formal verification to provide safety assurance for the decision-making component.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

2 A THREE-STAGE FORMAL APPROACH

To facilitate verifiable safe decision-making for autonomous systems, I have devised a three-stage formal approach consisting of formal specification, safe decision generation, and PCTL model checking. This formal approach is made possible through the utilization of four key components: a specification language named *vGOAL* that specifies autonomous decision-making mechanisms; an interpreter for *vGOAL* that automates safe decision-making generation; a translator that translates a given *vGOAL* specification to a PRISM model; and a PCTL model checker such as Storm [9] or PRISM [22] that verifies the soundness of the given *vGOAL* specification. My contributions are *vGOAL*, the interpreter of *vGOAL*, and the translator of *vGOAL*. The approach has been validated by a real-world autonomous logistic system consisting of three autonomous mobile robots. Three demos can be accessed at [26]: one demo for an error-free run, one demo for a run including a non-fatal error, and one demo for a run including a fatal error. The following briefly explains the key aspects of each stage.

2.1 Formal Specification: *vGOAL*

The formal specification requires a specification language that is expressive to specify autonomous decision-making mechanisms and suitable for formal verification. Additionally, it is advantageous to have a compatible interface with a widely used development framework for robotic applications.

Agent programming languages (APLs), including AgentSpeak [3], Jason [4], Gwendolen [11], and GOAL [18], have been extensively researched for programming autonomous agents for decades, rendering them well-suited for specifying autonomous decision-making mechanisms. GOAL shares many features with Belief-Desire-Intention (BDI) APLs, such as beliefs and goals, but it is primarily a rule-based APL [5]. Despite the potential benefits of APLs in the development of autonomous robotic systems, their research has not been widely used in the field. Integration with the Robot Operating System (ROS) may expand their applications to robotics, as ROS has become the de facto standard for developing robotic applications.

vGOAL is motivated by three primary considerations, which is a GOAL-based specification language that focuses exclusively on the internal logic reasoning mechanism of GOAL. First, GOAL is highly suitable for specifying autonomous decision-making, but many of its specifications are irrelevant to this domain, such as environment specifications. Second, the intrinsic logic-based nature of GOAL makes it highly suitable for formal verification. Third, GOAL has no build-in interface to ROS, which limits its applicability in robotic applications. Therefore, *vGOAL* can leverage the strengths of GOAL, formal verification, and ROS.

2.2 Safe Decision Generation: Interpreter

The agent-based decision-making component is implemented as the interpreter for *vGOAL*, which is integrated into ROS via rosbridge [7]. The motivation and the initial design of this stage were described in [27].

Figure 1 demonstrates how the interpreter generates safe decisions through interaction with ROS. ROS keeps sending real-time sensor information to the interpreter on a regular basis, e.g., every 100 milliseconds. The interpreter has three main components: a data processing component, a decision-making generation component, and a safety checking component. The interpreter generates decisions based on real-time information. Each generated decision will be checked if it violates any safety requirements. Only safe decisions can be sent to agents.

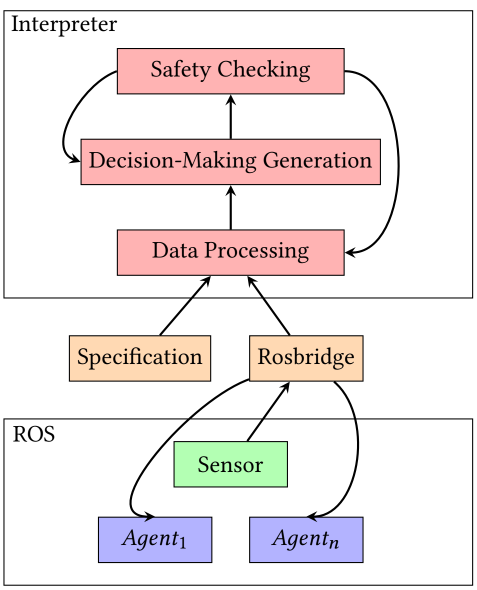

Figure 1: Safe Agent-Based Decision-Making Component

2.3 PCTL Model Checking: Translator

A *vGOAL* specification is considered sound when there is at least one feasible and safe plan available to achieve all goals. Formal verification is a compelling method for verifying the soundness of *vGOAL* specifications. Owing to the automated verification process, model checking is the most successful and influential verification method in verifying APLs, such as AgentSpeak(L) [2], Gwendolen, GOAL, SAAPL [24], and ORWELL [8] [13] [23] [28]. My approach encodes each state with its beliefs, allowing no additional computation for safety checking. This state encoding is easily expressible in the PRISM language, unlike in CTL model checkers such as SPIN [19] and NuSMV [6]. A finite transition system can be converted into a semantically equivalent finite discrete-time Markov chain (DTMC) except for transition probabilities. Moreover, the verification result of qualitative properties including safety and liveness properties in a finite DTMC is irrelevant to the transition probabilities [1]. Therefore, PCTL model checking is chosen to verify the soundness of *vGOAL* specifications.

Figure 2 illustrates the workflow of the automated PCTL model checking process. A translator was developed to convert a *vGOAL* specification into a PRISM model, with two components: transition system generation and PRISM model encoding. The translator generates the operational-semantically transition system from the *vGOAL* specification and encodes it as a semantically equivalent PRISM model except for transition probabilities. The translator and a PCTL model checker (Storm or PRISM) will automatically generate a PCTL model checking analysis from a *vGOAL* specification.

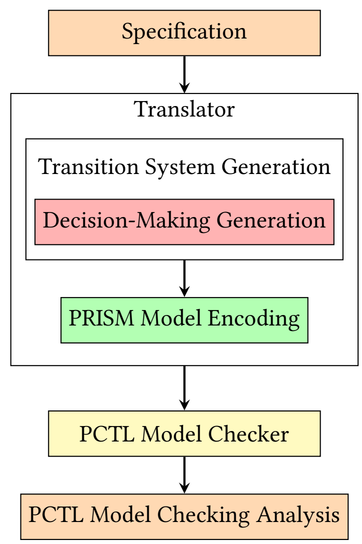

Figure 2: Automated PCTL Model Checking Process

3 FUTURE WORK

My future work includes two main directions. First, I plan to employ formal verification techniques, such as program verification, to prove the correctness of the implementation of the *vGOAL* interpreter. Second, I aim to investigate how a more sophisticated, safe, and intelligent motion planning component can be integrated into the agent-based decision-making module. Recent safe shielding techniques [17, 20, 21, 25] enable reinforcement learning-based control of autonomous systems in continuous state spaces while ensuring safety. It would be valuable to integrate the recent shielding techniques into the described agent-based decision-making component to provide safety assurance for autonomous systems.

ACKNOWLEDGEMENTS

This research is partially funded by the Research Fund KU Leuven.

REFERENCES

[1] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking*. MIT press.
[2] Rafael H Bordini, Michael Fisher, Carmen Pardavila, and Michael Wooldridge. 2003. Model checking agentspeak. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. 409–416.
[3] Rafael H Bordini and Jomi F Hübner. 2005. BDI agent programming in AgentSpeak using Jason. In *International workshop on computational logic in multi-agent systems*. Springer, 143–164.
[4] Rafael H Bordini, Jomi Fred Hübner, and Michael Wooldridge. 2007. *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons.
[5] Rafael C Cardoso and Angelo Ferrando. 2021. A review of agent-based programming for multi-agent systems. *Computers* 10, 2 (2021), 16.
[6] Alessandro Cimatti, Edmund Clarke, Fausto Giunchiglia, Marco Roveri, et al. 1999. NuSMV: A new symbolic model verifier. In *CAV*, Vol. 99. Citeseer, 495–499.
[7] Christopher Crick, Graylin Jay, Sarah Osentoski, Benjamin Pitzer, and Odest Chadwicke Jenkins. 2017. Rosbridge: Ros for non-ros users. In *Robotics Research: The 15th International Symposium ISRR*. Springer, 493–504.
[8] Mehdi Dastani, Nick AM Tinnemeier, and John-Jules Ch Meyer. 2009. A programming language for normative multi-agent systems. In *Handbook of Research on Multi-Agent Systems: semantics and dynamics of organizational models*. IGI Global, 397–417.
[9] Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. 2017. A storm is coming: A modern probabilistic model checker. In *International Conference on Computer Aided Verification*. Springer, 592–600.
[10] Louise Dennis and Michael Fisher. 2021. Verifiable autonomy and responsible robotics. *Software Engineering for Robotics* (2021), 189–217.
[11] Louise A Dennis and Berndt Farwer. 2008. Gwendolen: A BDI language for verifiable agents. In *Proceedings of the AISB 2008 Symposium on Logic and the Simulation of Interaction and Reasoning, Society for the Study of Artificial Intelligence and Simulation of Behaviour*. Citeseer, 16–23.
[12] Louise A Dennis, Michael Fisher, Nicholas K Lincoln, Alexei Lisitsa, and Sandor M Veres. 2016. Practical verification of decision-making in agent-based autonomous systems. *Automated Software Engineering* 23 (2016), 305–359.
[13] Louise A Dennis, Michael Fisher, Matthew P Webster, and Rafael H Bordini. 2012. Model checking agent programming languages. *Automated software engineering* 19, 1 (2012), 5–63.
[14] Michael Fisher, Emily Collins, Louise Dennis, Matt Luckcuck, Matt Webster, Mike Jump, Vincent Page, Charles Patchett, Fateme Dinmohammadi, David Flynn, et al. 2018. Verifiable self-certifying autonomous systems. In *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 341–348.
[15] Michael Fisher, Louise Dennis, and Matt Webster. 2013. Verifying autonomous systems. *Commun. ACM* 56, 9 (2013), 84–93.
[16] Michael Fisher, Viviana Mascardi, Kristin Yvonne Rozier, Bernd-Holger Schlingloff, Michael Winikoff, and Neil Yorke-Smith. 2021. Towards a framework for certification of reliable autonomous systems. *Autonomous Agents and Multi-Agent Systems* 35 (2021), 1–65.
[17] Andrew T Harris and Hanspeter Schaub. 2020. Spacecraft command and control with safety guarantees using shielded deep reinforcement learning. In *AIAA Scitech 2020 Forum*. 0386.
[18] Koen V Hindriks. 2009. Programming rational agents in GOAL. In *Multi-agent programming*. Springer, 119–157.
[19] Gerard J. Holzmann. 1997. The model checker SPIN. *IEEE Transactions on software engineering* 23, 5 (1997), 279–295.
[20] Nathan Hunt, Nathan Fulton, Sara Magliacane, Trong Nghia Hoang, Subhro Das, and Armando Solar-Lezama. 2021. Verifiably safe exploration for end-to-end reinforcement learning. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*. 1–11.
[21] Nils Jansen, Bettina Könighofer, JSL Junges, AC Serban, and Roderick Bloem. 2020. Safe reinforcement learning using probabilistic shields. (2020).
[22] Marta Kwiatkowska, Gethin Norman, and David Parker. 2002. PRISM: Probabilistic symbolic model checker. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*. Springer, 200–204.
[23] Gerhard Weiss. 2013. *Multiagent Systems*. The MIT Press. 462 pages.
[24] Michael Winikoff. 2007. Implementing commitment-based interactions. In *Proceedings of the 6th international joint conference on Autonomous agents and multi-agent systems*. 1–8.
[25] Wen-Chi Yang, Giuseppe Marra, Gavin Rens, and Luc De Raedt. 2023. Safe Reinforcement Learning via Probabilistic Logic Shields. *arXiv preprint arXiv:2303.03226* (2023).
[26] Yi Yang. 2022. Demo Video. https://kuleuven-my.sharepoint.com/:f:/g/personal/yi_yang_kuleuven_be/Er11lD515VhGjoMy59zHHHMBHr6r8aTID_sFK1MEQPI3Lw?e=KG9lCo.
[27] Yi Yang and Tom Holvoet. 2022. Generating Safe Autonomous Decision-Making in ROS. In *Fourth Workshop on Formal Methods for Autonomous Systems*, Vol. 371. Open Publishing Association, 184–192. https://doi.org/10.4204/EPTCS.371.13
[28] Yi Yang and Tom Holvoet. 2022. Making Model Checking Feasible for GOAL. In *10th International Workshop on Engineering Multi-Agent Systems*.