Dr. Radu Calinescu[1] Dr. Colin Paterson[1]
Dr. Alec Banks[2]
Dr. Daniel Kudenko[3]

Joshua Paul Riley[1]

[1] Department of Computer Science, University of York, York, UK.
[2] Defence Science and Technology Laboratory, UK.
[3] L3S Research Centre, Leibniz University, Hanover, Germany

Assured Multi-Agent Reinforcement Learning

Putting the *Trust* in Trustworthy Robotic Teams

UNIVERSITY of York

ASSURING AUTONOMY INTERNATIONAL PROGRAMME

[dstl]

1. What Assured Multi-Agent Reinforcement Learning can do for you!

Multi-robot systems have been proposed for a myriad of exciting and game-changing scenarios*. Many of which will remove humans from potential harm and increase work performance and quality of life.

Without assurances that these systems will behave safely and that their safe behaviour doesn't compromise mission objectives, their use will remain significantly limited.

Assured Multi-Agent Reinforcement Learning supplies these assurances through quantitatively verified constraints that provide formal guarantees on both safety and functionality—helping to bridge the gap between contained system use and real-world use.

Exciting and Game-Changing Scenarios*: Search and Rescue, Hospital Assistance, Planetary Exploration, Nuclear Power Plant Operations, Care Home Assistance, Security Operations, Military Operations, Agriculture.

Brinkmann, W., et al., 2019. Space robotic systems

Alonso-Mora, J., et al., 2017. Multi-robot formation control.

2. Background

Issues of Multi-Agent Reinforcement Learning (MARL)

Reinforcement learning (RL) is a machine learning technique with roots in behavioural psychology. An agent will receive numerical rewards or punishments when it takes actions within an environment based on the utility of said action.

RL uses two types of behavioural selection methods: exploitation, where the agent selects the most useful action, and exploration, which selects an action at random. It is in this stochastic process where safety issues occur due to unpredictability.

Abstract Markov Decision Process (AMDP)

Working with MDPs that include all states and state transitions can easily become conflated, especially in large environments. An MDP can be abstracted to enable MDPs to be used efficiently and be eligible for analysis.

When abstracting an MDP, identifying important states and states with commonality is required. Important states will be represented within the AMDP, but the states that share great commonality can be grouped together and represented with a single abstracted state. Such as, all states within a room can be abstracted into a single state that represents the entirety of the room.

Quantitative Verification (QV)

QV is a technique that allows the analysis of quantitative aspects of a system, such as reliability, safety, performance etc. It makes use of formal methods to analyse mathematical models such as an MDP. Using QV, for example, it is possible to determine the probability of a system completing a goal within a certain amount of actions. Some commonly used probabilistic model checking tools are PRISM and STORM.

PRISM allows us to construct an MDP using the PRISM coding language and describe requirements using probabilistic computational tree logic (PCTL). An example of such a description can be seen here: **P**>=1 [**F** "terminate"] meaning "the algorithm eventually terminates successfully with probability 1".

3. Assured Multi-Agent Reinforcement Learning (AMARL)

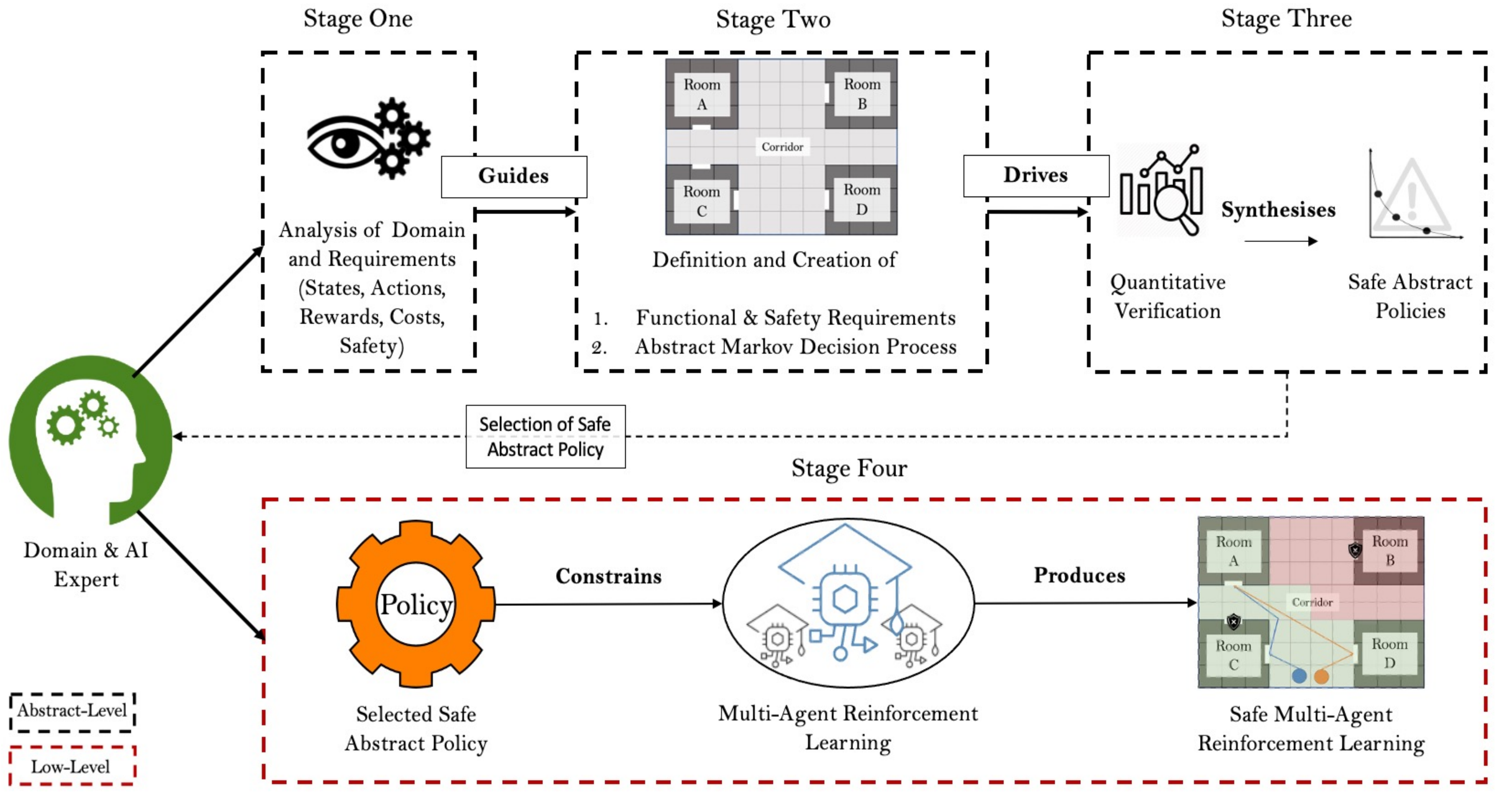
Description

AMARL is an approach that delivers safe multi-agent reinforcement learning policies that comply with formal guarantees on functional and safety objectives, negating the stochastic issues of RL.

These formal guarantees are supplied through QV, which synthesis safe abstract policies used to constrain agents. This synthesis allows a level of trust to be placed on a system, which was not present before.

AMARL is unique as it allows agents to enter into risky environments and perform risky actions, allowing more flexible use of MARL but with confidence they will meet safety requirements.

It is designed to be used on a broad range of issues and systems by utilising a plug-in style for a plethora of needs.



Stage One

Analysis of Domain and Requirements (States, Actions, Rewards, Costs, Safety)

Guides

Stage Two

Definition and Creation of

1. Functional & Safety Requirements
2. Abstract Markov Decision Process

Drives

Stage Three

Quantitative Verification

Synthesises

Safe Abstract Policies

Selection of Safe Abstract Policy

Domain & AI Expert

Stage Four

Selected Safe Abstract Policy

Constrains

Multi-Agent Reinforcement Learning

Produces

Safe Multi-Agent Reinforcement Learning

Abstract-Level

Low-Level

The Stages

1. Analysis of the problem domain and the requirements of the system. Identify key features in preparation for stage two.

2. Creation of an AMDP to allow crucial information to be captured while allowing efficient QV. Formally define functional and safety requirements using PCTL.

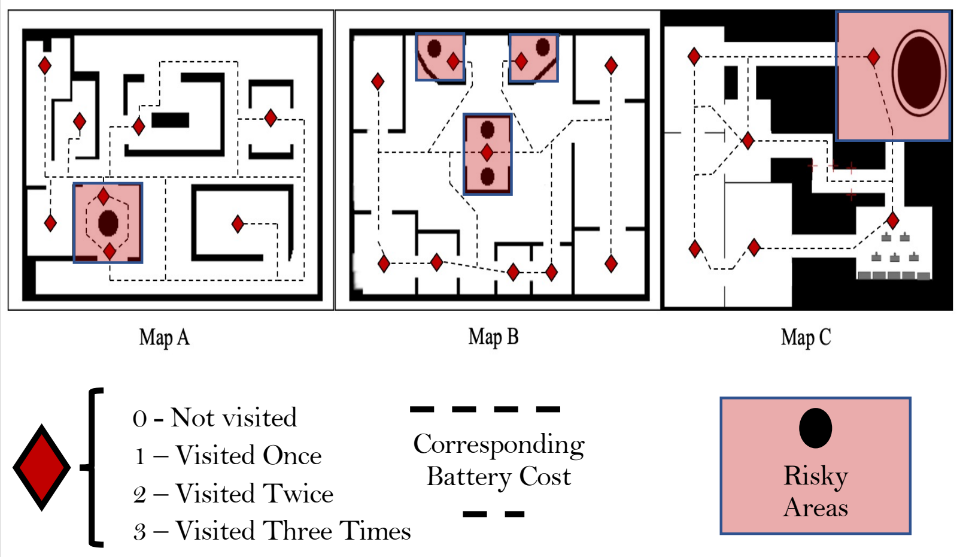
3. QV, through probabilistic model checking, synthesises safe abstract policies. These abstract policies will provide formal guarantees of safety and functionality.

4. Safe abstract policy selection to constrain the MARL agents from preforming unwanted behaviour, which produces safe policies.

4. Domain Examples

We ran our experiments on three 'nuclear power plant' inspired domains, with varying system sizes, system types, and algorithms to showcase the plugin nature of our AMARL approach.

In Maps A, B, and C, robotic agents must visit all patrol points 'diamonds' at least three times between themselves while using as few actions as they can to preserve battery and limiting their time in risky areas.



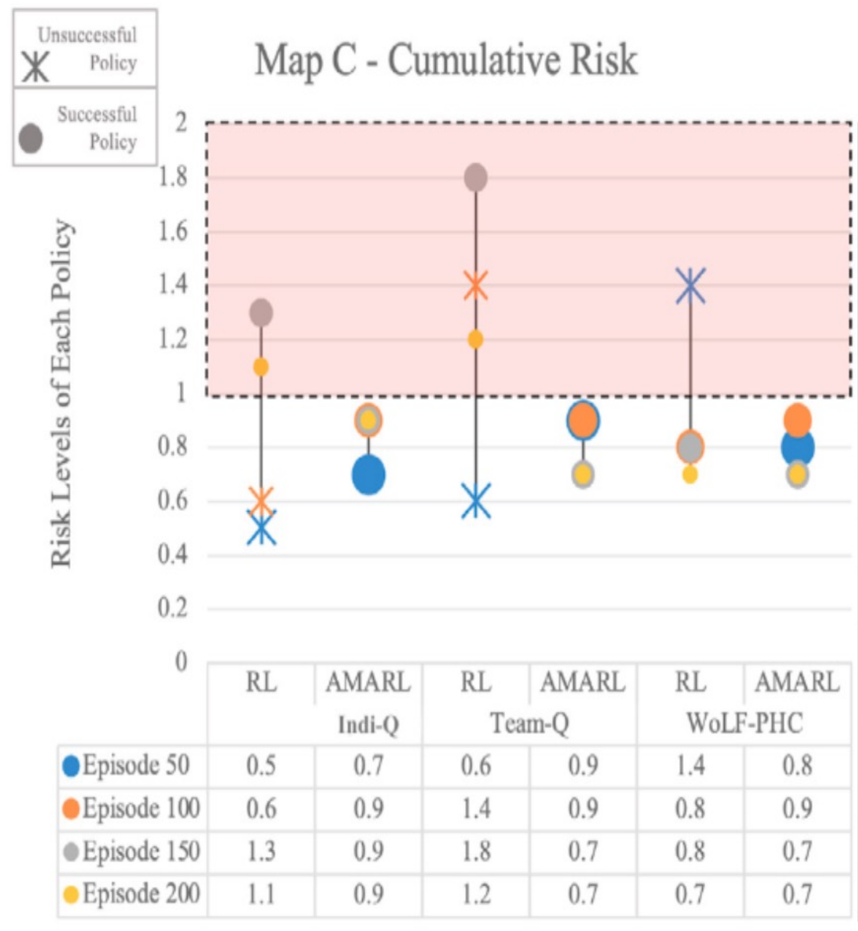
We frame these problems as an inspection problem, where agents must inspect sensitive equipment over time. By using robotic teams instead of humans, we show the potential of our research leading to safer conditions for humans.

5. Results From One Set Of Experiments

Running three separate MARL algorithms with differing behaviours, we analyse the battery consumption and the cumulative risk of learning run over episodes.

The unacceptable level of cumulative risk is shown as a light red square. As can been seen, the algorithms constrained using our approach never entered into this square and, in general, were much more predictable than unconstrained learning.

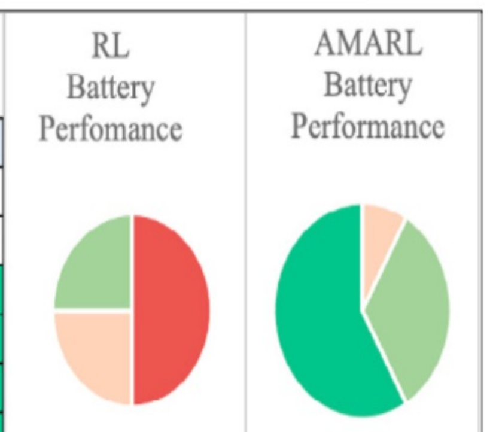
Due to the nature of the constrained approach limiting the state-space, the battery performance was much more efficient using our constrained approach, as seen by the lack of red in AMARLs performance seen in the battery graph.



	RL	AMARL	RL	AMARL	RL	AMARL
	Indi-Q		Team-Q		WoLF-PHC	
● Episode 50	0.5	0.7	0.6	0.9	1.4	0.8
● Episode 100	0.6	0.9	1.4	0.9	0.8	0.9
● Episode 150	1.3	0.9	1.8	0.7	0.8	0.7
● Episode 200	1.1	0.9	1.2	0.7	0.7	0.7

Failed	Passable	Efficient	Highly Efficient

	Indi-Q			
	RL		AMARL	
	Robot-1	Robot-2	Robot-1	Robot-2
Episode 50	0	13	63	62
Episode 100	0	27	68	63
Episode 150	13	48	68	66
Episode 200	16	48	68	66



	Team-Q				WoLF-PHC			
	RL		AMARL		RL		AMARL	
	Robot-1	Robot-2	Robot-1	Robot-2	Robot-1	Robot-2	Robot-1	Robot-2
Episode 50	27	0	48	41	0	31	40	37
Episode 100	5	0	49	48	38	34	54	33
Episode 150	26	44	49	45	41	38	54	50
Episode 200	39	49	53	44	43	38	54	50

6. Conclusion

From our current experiments, we can clearly see a trend forming. This trend supports our approach to satisfy the demand for safety and assurances and establish these assurances with formal methods, with homogenous and heterogeneous robotic teams of differing sizes.

Our work, being the first approach to apply quantitative verification for MARL constraint in this fashion, helps establish a bridge between the academic use of MARL and its use in safety-critical scenarios.

To view more see: *Utilising Assured Multi-Agent Reinforcement Learning within Safety-Critical Scenarios, 2021.*